Ref: MITM/CE/Scheme/2023-24/002

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Autonomous Institution Affiliated to VTU

Competency Based Syllabus (CBS)

for

Computer Engineering

(Under Outcome Based Education (OBE) and

Choice-Based Credit System (CBCS))

Offered from 3rdto 4th Semesters of Study

in

Partial Fulfillment for the Award of Bachelor's Degree in

Computer Engineering

2023 Scheme

Scheme Effective from the academic year 2023-24

General Contents of Competency Based Syllabus Document

Index	Description
1	Prerequisites
2	Competencies
3	Syllabus
4	Syllabus Timeline
5	Teaching-Learning Process Strategies
6	Assessment Details
7	Learning Objectives
8	Course Outcomes and Mapping with POs/ PSOs
9	Assessment Plan
10	Future with this Subject

3rd Semester	Basic Science Course (BS) BIOLOGY FOR ENGINEERS	M23BBIOK301

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Knowledge of Biology	A basic understanding of high school-level biology concepts can be beneficial. This includes knowledge of cell structure, basic physiology, and fundamental biological processes.
2.	Basic Knowledge of Biochemistry	Familiarity with major biological molecules and their application (carbohydrates, proteins, lipids, nucleic acids, vitamins, enzymes, and hormones).
3.	Basic Knowledge of Mathematics and Physics	Understanding the human anatomy and physiological systems in comparison with bioengineering principles.
4.	Basic Concepts of Design	Understanding of basic design and system thinking, which will help in bioengineering design and nature-bioinspired materials and mechanisms?
5.	Engineering Fundamentals	Ability to analyze and apply basic engineering principles to solve biological problems.

2. Competencies

S/L	Competency	KSA Description
1.	**Cell Structure and Function**	**Knowledge:** • Understand the fundamentals of Cell Biology **Skills:** • Efficient file manipulation, text pro. **Attitudes:** • Appreciate the complexity and diversity of cellular structures. • Demonstrate an interest in how biomolecules contribute to life processes.
2.	**Biomolecules**	**Knowledge:** • Understanding the applications of Biomolecules. **Skills:** • Analyze and apply the knowledge of Biomolecules. **Attitudes:** • Demonstrate an interest in how biomolecules contribute to life processes.
3.	**Anatomical Principles for Bioengineering Design**	**Knowledge:** • Understanding the human anatomical administration. **Skills:** • Apply knowledge of human anatomy to bioengineering projects **Attitudes:** • Appreciate the ingenuity of biological systems and their engineering potential. • Exhibit creativity in applying anatomical principles to engineering problems.
4.	**Nature-Bioinspired Materials and Mechanisms**	**Knowledge:** • Comprehend the principles behind bioinspired materials and mechanisms **Skills:** • Analyze and apply knowledge of natural principles to design innovative materials and systems. **Attitudes:** • Demonstrate curiosity about how natural systems work and their potential applications. • Exhibit a proactive approach to learning from nature to solve engineering challenges.

5.	**Trends In Bioengineering**	**Knowledge:** • Comprehend the principles and applications behind bioengineering. **Skills:** • Analyze and apply knowledge of bioengineering principles to understand various environmental and industrial contexts. **Attitudes:** • Demonstrate curiosity about how natural systems work and their potential applications. • Exhibit a proactive approach to learning from nature to solve engineering challenges.

3.Syllabus

BIOLOGY FOR ENGINEERS
SEMESTER –IV

Course Code	**M23BBIOK301**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(1:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**15 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**01**

Course objectives:

- To acquaint the students with fundamental biological principles and their application to bioengineering.
- To enable the students to understand the bio-design principles to create novel devices and structures.
- To show the students how biological systems can be re-designed as substitute products for natural systems.
- To encourage students to create an interdisciplinary view of biological engineering.

MODULE - 1 (3 Hours)

CELL BIOLOGY

Introduction to cell (Types, structure, and major functions of Cells and Cell Organelles) Stem cells and their application. Biomolecules: Properties and functions of Carbohydrates, Nucleic acids, Proteins, Lipids, Enzymes, Vitamins, and Hormones.

MODULE 2 (3 Hours)

BIOMOLECULES AND THEIR APPLICATION

Carbohydrates as Cellulose-based water filters, PHA and PLA as Bioplastics, Nucleic acids in Vaccines and Diagnosis, Proteins in food production (Plant-based protein, Whey protein, and Meat analogs), Lipids as biodiesel, and cleaning agents/detergents, Enzymes in Biosensors fabrication, Food processing, Detergent formulation, and Textile processing.

MODULE 3 (3 Hours)

ADAPTATION OF ANATOMICAL PRINCIPLES FOR BIOENGINEERING DESIGN

Brain as a CPU System. Eye as a Camera System. Heart as a Pump System. Lungs as Purification System. Kidney as a Filtration System.

MODULE 4 (3 Hours)

NATURE-BIOINSPIRED MATERIALS AND MECHANISMS

Echolocation, Photosynthesis. Bird Flying, Lotus Leaf Effect, Plant Burrs, Sharkskin, Kingfisher Beak. Human Blood Substitutes - Hemoglobin-Based Oxygen Carriers (Hbocs) and Perfluorocarbons (Pfcs).

MODULE 5 (3 Hours)

TRENDS IN BIOENGINEERING:

Scaffolds In Muscular, Skeletal Systems and Tissue Engineering, Bioprinting Techniques and Materials. Electrical Tongue and Electrical Nose in Food Science, DNA Origami and Biocomputing, Bioimaging, and Artificial Intelligence for Disease Diagnosis. Bioconcrete. Bioremediation. Biomining.

Text Book(s)

1. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.
2. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.

Reference Books

1. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
2. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
3. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
4. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
5. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
6. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
7. Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N GeethaA C Udayashankar Lambert Academic Publishing, 2019.
8. 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
9. Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016.

Web links and Video Lectures (e-Resources):

https://nptel.ac.in/courses/121106008

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Cell Biology	**Competency:** Understand cell biology and biomolecule functions. **Knowledge:** Understand and describe the types, structures, and major functions of cells and their organelles. **Skills:** Identify and describe the importance of the cell
2	Week 3-5: Biomolecules and their Applications	**Competency:** Apply knowledge of biomolecules to real-world applications **Knowledge:** Understand the properties and functions of key biomolecules: carbohydrates, nucleic acids, proteins, lipids, enzymes, vitamins, and hormones. **Skills:** Analyze the role of various biomolecules
3	Week 6-8: Adaptation Of Anatomical Principles for Bioengineering Design	**Competency:** Translate anatomical principles into bioengineering designs. **Knowledge:** Understand anatomical principles to bioengineering designs, drawing analogies such as the brain as a CPU system, the eye as a camera system, the heart as a pump system, the lungs as a purification system, and the kidney as a filtration system. **Skills:** Analyze the role of human anatomy with bioengineering principles
4	Week 9-10: Nature-Bioinspired Materials and Mechanisms	**Competency:** Utilize bioinspired materials and mechanisms. **Knowledge:** Understand and explain natural bio-inspired mechanisms. **Skills:** Apply knowledge of natural principles to design innovative materials and systems.
5	Week 11-12: Trends in Bioengineering	**Competency:** Keep abreast of current trends and technologies in bioengineering. **Knowledge:** Understand the principles and applications of various bioengineering mechanisms. **Skills:** Apply knowledge of biology in engineering trends in various filed of science.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Lecture Method**	Explanation via real-life problems, situation modeling, deliberation of solutions, hands-on sessions, reflective and questioning /inquiry-based teaching.

2	**Live Demonstration**	Instructions with interactions in classroom lectures (physical/hybrid).
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**ICT Tools**	Use of ICT tools, including YouTube videos, related MOOCs, AR/VR/MR tools.
5	**Problem-Based Learning (PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Multiple Representations**	Introduce topics in various representations to reinforce competencies
7	**Gamification Tools**	Use of gamification tools (in both physical/hybrid classes) for creative learning outcomes
8	**Student Seminars**	Solo, group /oral presentations.
9	**Model Making**	Demonstration using working models.

6. Assessment Details (both CIE and SEE)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks (A+B)			**50**	**20**

The CIE question paper shall have MCQ set for 25 questions, each carrying one mark.
Average internal assessment shall be the average of the 2 test marks conducted.
Semester End Exam (SEE)
The SEE question paper shall have MCQ set for 50 questions, each carrying one mark. The time duration for SEE is one hour

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the fundamentals of Cell	Students will understand the types, structures, and functions of cells and their organelles.
2	Analyze the application of Biomolecules	Students will be able to analyze the practical applications of carbohydrates, proteins, nucleic acids, lipids, and enzymes.
3	Bioengineered System Models	Students will be able to translate anatomical principles into bioengineering designs.
4	Bioinspired Mechanism Models	Students will be able to explore and replicate nature-inspired materials and mechanisms.
5	Emerging Bioengineering Technology Models	Students will be able to understand and demonstrate the latest trends in bioengineering.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BBIOK301.1	Elucidate the fundamentals of biological concepts employing pertinent health, and engineering applications.
M23BBIOK301.2	Assess the biological ideologies for the design and development of novel bioengineering solutions.
M23BBIOK301.3	Substantiate and apply the ideologies amid nature-inspired biomimetics perceptions for explicit engineering solutions.
M23BBIOK301.4	Exploring innovative bio-based solutions for relevant biological complications.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BBIOK301.1	3	-	-	-	-	-	3	-	-	-	-	3	-	-
M23BBIOK301.2	3	-	3	-	-	3	-	-	-	-	-	3	-	-
M23BBIOK301.3	3	3	3	-	-	-	3	-	-	-	-	3	-	-
M23BBIOK301.4	3	-	3	-	3	-	3	-	-	-	-	-	-	-
M23BBIOK301	3	3	3		3	3	3	-	-	-	-	3	-	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2		**10**			**10**
Module 3			**10**		**10**
Module 4				**10**	**10**
Module 5	**3**	**2**	**3**	**2**	**10**
Total	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2		**20**			**20**
Module 3			**20**		**20**
Module 4				**20**	**20**
Module 5	**5**	**5**	**5**	**5**	**20**
Total	**25**	**25**	**25**	**25**	**100**

Conditions for SEE Paper Setting:
Each module of the SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

By equipping students with the knowledge, skills, and attitudes necessary to excel in bioengineering, the "Biology for Engineers" course prepares them for a future where they can make significant contributions to healthcare, environmental sustainability, and numerous other fields where biology and engineering intersect.

➢ **Future Trends in Bioengineering**

1. **Personalized Medicine:** Understanding genetics and molecular biology to design personalized medical treatments.
    - Applications: Developing patient-specific drugs, gene therapy, and personalized treatment plans based on individual genetic profiles.
2. **Regenerative Medicine and Tissue Engineering:** Studying stem cells, scaffolding materials, and growth factors.
    - Applications: Creating artificial organs, repairing damaged tissues, and developing bioengineered skin for burn victims.
3. **Bioprinting:** Learning about 3D printing techniques and biomaterials.

- Applications: Printing tissues and organs, developing complex tissue structures for research and therapeutic use.

4. **Synthetic Biology:** Engineering biological systems for new functions.
   Applications: Designing microorganisms to produce biofuels, clean pollutants, or synthesize pharmaceuticals.
5. **Biomedical Imaging and Diagnostics**: Understanding imaging technologies and diagnostic tools.
   - Applications: Advancing MRI, CT scans, and other imaging technologies to improve diagnostic accuracy and patient outcomes.
6. **Wearable Health Technologies:** Integrating biology with electronics and materials science.
   - Applications: Developing wearable devices that monitor health metrics, detect diseases early, and provide real-time health data to patients and doctors.
7. **Artificial Intelligence in Healthcare:** Combining biology with data science and machine learning.
   - Applications: Using AI to analyze complex biological data, predict disease outbreaks, and personalize medical treatments.
8. **Environmental Bioengineering:** Applying biological principles to environmental challenges.
   - Applications: Bioremediation, biomining, and developing sustainable agricultural practices.

➢ **Career Paths for Bioengineers**

1. **Biomedical Engineer**:
   - Role: Design and develop medical devices, prosthetics, and diagnostic equipment.
   - Skills: Combining engineering principles with biological knowledge to solve medical problems.
2. **Clinical Research Scientist**:
   - Role: Conduct research to improve medical technologies and treatment methods.
   - Skills: Applying biological and engineering expertise to clinical trials and laboratory research.
3. **Biotech Product Manager**:
   - Role: Oversee the development and marketing of biotech products.
   - Skills: Understanding both the technical aspects of bioengineering and the commercial landscape.
4. **Regenerative Medicine Specialist**:
   - Role: Focus on developing therapies that regenerate damaged tissues and organs.
   - Skills: Combining knowledge of cell biology, biomaterials, and clinical applications.
5. **Environmental Engineer**:
   - Role: Develop solutions for environmental problems using biological principles.
   - Skills: Applying bioengineering techniques to waste management, pollution control, and sustainable development.
6. **Bioinformatics Specialist**:
   - Role: Analyze biological data using computational tools.
   - Skills: Merging biology with computer science to interpret complex data sets and develop new algorithms for biological research.
7. **Bioprocess Engineer**:
   - Role: Design and optimize processes for producing biological products.
   - Skills: Understanding both the biological and engineering aspects of bioproduction, including scaling up processes from lab to industry.

3rd Semester	**Professional Course (PC)** **LOGIC DESIGN AND COMPUTER ORGANIZATION**	**M23BCS302**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Digital Logic	Understanding of Boolean algebra. Familiarity with fundamental digital logic concepts such as gates, flip-flops, and combinational and sequential circuits.
2.	Electronic Circuits	Knowledge of basic electronic components and circuits. Understanding of voltage, current, and their behavior in electronic circuits.
3.	Mathematics	Proficiency in algebra for Boolean expression simplification using K-map techniques
4.	Programming Concepts (for Verilog	Basic understanding of programming concepts
5.	Logic Circuit Analysis	Ability to analyze and design logic circuits, including combinational and sequential circuits
6.	Basic Understanding of Digital Logic	Knowledge of basic digital logic gates (AND, OR, NOT, etc.). Understanding of Boolean algebra and logic simplification techniques
7.	Fundamental Electronics Knowledge	Knowledge of basic digital logic gates (AND, OR, NOT, etc.). Understanding of Boolean algebra and logic simplification techniques
8.	Fundamental Electronics Knowledge	Basic understanding of electronic components, circuits, and their behavior. Knowledge of binary number system and arithmetic.
9.	Programming Fundamentals	Basic programming skills, as the course involves digital system design
10.	Previous Coursework	Completion of introductory courses in Basic electronics or a related field

2. Competencies

S/L	Competency	KSA Description
1.	**Boolean Expression Simplification**	**Knowledge:** Understanding of Boolean algebra principles. Knowledge of Karnaugh maps and Quine-McCluskey minimization techniques. **Skills:** Ability to apply K-map techniques for Boolean expression simplification. Proficiency in utilizing Quine-McCluskey minimization techniques. **Attitudes:** Appreciation for the importance of logical simplification in digital system design.
2.	**Combinational Logic Circuits**	**Knowledge:** Understanding of combinational logic principles and canonical forms. **Skills:** Designing combinational logic circuits based on specifications. Analyzing and evaluating the performance of combinational logic circuits. **Attitudes:** Appreciation for the role of combinational logic in digital systems.
3.	**Sequential Logic Circuits**	**Knowledge:** Understanding of flip-flops, registers, and sequential logic principles. **Skills:** Designing sequential logic circuits with flip-flops. Optimizing the behavior of sequential circuits. **Attitudes:** Valuing the importance of sequential logic in digital system functionality
4.	**Assembly language**	**Knowledge:** Understanding the structure of assembly language modules. Knowledge of assembly language operators and data types.

		Skills: Applying **ASP** for digital system design. Describing digital systems using assembly programming language. **Attitudes:** Openness to learning and using hardware description languages for design.
5.	**Logic Design with MSI Components and PLDs**	**Knowledge:** Understanding of MSI components and PLDs. **Skills:** Implementing binary adders, subtractors, comparators, and multiplexers. Utilizing programmable logic devices (PLDs) in logic design. **Attitudes:** Appreciation for the versatility of MSI components and PLDs in digital logic design.
6.	**Flip-Flops and its Applications**	**Knowledge:** Understanding the characteristics of flip-flops. **Skills:** Designing and analyzing binary ripple counters and synchronous binary counters. Implementing mod-n counters using different flip-flops. **Attitudes:** Recognizing the significance of flip-flops in sequential logic circuits
7.	**Computer structure and machine instructions**	**Knowledge:** Basic operational unit of computer system Branch Instruction and Addressing modes **Skills:** Writing Assembly language program Encoding the Machine Instructions **Attitudes:** Openness to learning and using assembly language program for design
8.	**Input/Output and memory unit**	**Knowledge:** Functions of I/O devices and memory units Study of Interrupts and Exceptions **Skills:** Measuring the performance of the computer system How to prioritize the Interrupt **Attitudes:** Valuing the importance of different input/output and memory units in digital system functionality

3. Syllabus

<table>
<tr><td colspan="4"><b>LOGIC DESIGN AND COMPUTER ORGANIZATION
SEMESTER – III</b></td></tr>
<tr><td>Course Code</td><td><b>M23BCS302</b></td><td>CIE Marks</td><td><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week(L: T: P: S)</td><td><b>(3:0:0:0)</b></td><td>SEE Marks</td><td><b>50</b></td></tr>
<tr><td>Total Number of Lecture Hours</td><td><b>40 Hours</b></td><td>Total Marks</td><td><b>100</b></td></tr>
<tr><td>Credits</td><td><b>03</b></td><td>Exam Hours</td><td><b>03</b></td></tr>
<tr><td colspan="4"><b>Course objectives:</b> This course will enable students to:
1. Explain the functionality of binary logic system
2. Illustrate the working of different combinational circuits
3. Illustrate the working of different flipflops, registers and counters
4. Realize the basic concepts of computer system
5. Illustrate the working of I/O and memory units</td></tr>
<tr><td colspan="4"><b>Module -1</b></td></tr>
<tr><td colspan="4"><b>Introduction to Digital Logic</b>: Binary Logic, Boolean algebra, Canonical and standard forms, digital logic gates, IC digital logic families, simplification of Boolean expressions using k-map techniques up to 4 variables (including don’t-care condition), NAND-NOR implementation, Quine-McClusky method.
Text1: 2.1 to 2.8, 3.1 to 3.3, 3.5, 3.6, 3.8, 3.9</td></tr>
<tr><td colspan="4"><b>Module -2</b></td></tr>
<tr><td colspan="4"><b>Combinational logic and its applications:</b> Introduction<b>,</b> Design procedure, adders, subtractors, Code conversions, , multilevel NAND and NOR, Exclusive OR and Equivalence function, binary adder and subtractor circuits, decimal adder, magnitude comparator, decoder, encoder, multiplexer, PROM, PLA
Text1: 4.1 to 4.9, 5.1 to 5.8</td></tr>
<tr><td colspan="4"><b>Module -3</b></td></tr>
</table>

Sequential logic and its applications: introduction, flip flops, SR latch, gated SR latch, D Latch, Edge triggered D flip flop, JK, T, SR, Master slave flip flop, Shift registers, ripple counter, analysis of clocked sequential circuits.
Text1: 6.1 to 6.9, 7.1 to 7.5

Module -4

Fundamentals of computer structure and machine instructions: Basic Operational Units, Bus Structure, Performance, Memory location and addresses, Memory Operations, Instruction and Instructions sequencing, addressing modes, assembly language,
Text 2: 1.3, 1.4, 1.6(1.6:1, 3, 4, 7), 2.2, 2.3, 2.4, 2.5, 2.6

Module -5

Organization of Input/output and memory unit: Accessing I/O devices**,** Interrupts**,** Direct memory access, I/O interfaces (PCI, USB, SCSI), RAM memories, ROM memories, Memory hierarchy in cost, speed and size, cache memory and its mapping techniques, pipelining technique.
Text2: 4.1, 4.2(exept4.4.6), 4.4, 4.7, 5.2, 5.3, 5.4, 5.5(5.5.1, 5.5.2)

Text Books:
1.M. Morris Mano, Digital logic and computer desig, Pearson, 2011
2. Carl Hamachar, Zvoko Vranesic, Safwat Zaky, Computer organization, 5th edition, Tata McGraw Hill, 2002

Reference Books:
1. Donald D Givonne, Digital principles and design, Tata McGraw Hill, 2002.
2. Shuangbao Paul wang, Robert S. Ledley Computer Architecture and Security, 2013

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction to Digital Logic	**Competency**: Boolean Expression Simplification **Knowledge**: Boolean algebra principles, Karnaugh maps, and Quine-McCluskey techniques. **Skills**: Applying K-map techniques, utilizing Quine-McCluskey techniques.
2	Week 3-4: Combinational logic and its applications	**Competency**: Logic Design with MSI Components and PLDs **Knowledge**: MSI component functionalities, Programmable Logic Devices (PLDs). **Skills:** Implementing binary adders, subtractors, and combinational circuits
3	Week 5-6: Sequential logic and its applications	**Competency:** Sequential Logic Circuits **Knowledge**: Sequential logic principles, Flip-flop characteristics, and counting circuits. **Skills**: Designing and optimizing sequential logic circuits
4	Week 7-8: Fundamentals of computer structure and machine instructions	**Competency:** Basic Computer structure **Knowledge:** Basic operations of computer system **Skills:** Writing Assembly language program, Encoding Machine instructions
5	Week 9-10: Organization of Input/Output and memory unit	**Competency:** Study of Input/output devices and Memory unit **Knowledge**: Structural organization of I/O device and memory unit **Skills**: Measuring the performance of the computer system

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Collaborative Learning	Encourage collaborative learning for improved competency application.
3	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.

4	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
5	Multiple Representations	Introduce topics in various representations to reinforce competencies
6	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
7	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
8	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

CIE methods /question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	**Learning Objectives**	**Description**
1	Understanding Digital Design Fundamentals	Students will grasp the fundamental concepts of digital logic design, including Boolean algebra, logic gates, combinational and sequential circuits, and finite state machines
2	Designing Combinational and Sequential Circuits	Students will learn to design and implement combinational circuits such as adders, multiplexers, and decoders, as well as sequential circuits such as flip-flops, registers, and counters using Verilog
3	Project-Based Learning	Through hands-on projects, students will apply their knowledge of digital design to design, implement, simulate, and verify complex digital systems, reinforcing their understanding of theoretical concepts
4	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
5	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS302.1	Apply the reduction techniques to solve the boolean expressions.
M23BCS302.2	Construct various combinational circuits using simplification techniques.
M23BCS302.3	Develop flip-flop applications for sequential logic circuits.
M23BCS302.4	Apply the fundamentals of addressing modes to develop ALP.
M23BCS302.5	Utilize the concepts of I/O and memory units to measure the performance.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS302.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS302.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS302.3	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS302.4	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS302.5	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS302	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Logic Design and Computer Organization" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

- **Advanced Digital Design Courses:** The knowledge gained in this course, covering principles of combinational and sequential logic circuits serves as a prerequisite for more advanced courses in digital design. Students can delvelop deeper into topics such as FPGA-based design, advanced digital system architectures, and high-performance digital systems.

- **Embedded Systems:** Understanding application of Logic design and Computer Organization is crucial for students pursuing courses related to embedded systems (Iot projects).
- **VLSI Design:** The course provides a solid foundation for students interested in pursuing VLSI (Very Large Scale Integration) design courses. The principles of combinational and sequential logic, are essential for designing complex integrated circuits.
- **Digital Signal Processing (DSP):** Students specializing in DSP benefit from the course by gaining insights into the fundamentals of logic design and computer organization. This knowledge becomes instrumental when working on DSP algorithms, hardware implementations, and the design of specialized digital signal processors.
- **Computer Architecture and Organization:** The course contributes to the understanding of computer organization and architecture. Concepts such as binary adders, decoders, and multiplexers are foundational to the study of computer architecture, providing insights into how digital systems are organized and interconnected.
- **Advanced Programming Courses:** Students pursuing courses in advanced programming, especially those related to hardware programming or system-level programming, concepts of digital circuits and memories can leverage their knowledge when developing g on software-hardware co-design projects.
- **Project Work and Research:** The hands-on experience gained through programming assignments, problem-solving, and project work in digital circuits and IoT based embedded design, to enhance their skills needed to upgrade their knowledge.
- **Industry Applications:** The course provides practical skills that are directly applicable in industries related to digital system design, VLSI, embedded systems, and more. Graduates are well-prepared to contribute to industries by applying the concepts of digital circuits

3rd Semester	Integrated Professional Course (IPC) OPERATING SYSTEMS	M23BCS303

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic programming skills	Understanding of programming concepts and experience in a programming language such as C, C++, Java, or Python
2	Computer architecture	Basic understanding of computer organization and architecture, including concepts like memory hierarchy, CPU operations, and input/output systems.
3	Data structures and algorithms	Familiarity with fundamental data structures (arrays, linked lists, trees, etc.) and algorithms (sorting, searching, etc.).
4	Computer Organization	Knowledge of how hardware components interact at a low level is helpful. This includes concepts like machine instructions, addressing modes, memory management, and I/O operations.
5	Databases	Basic understanding of databases and file management systems is helpful as operating systems often interact with databases for storing and retrieving data.

2. Competencies

S/L	Competency	KSA Description
1	Operating System Fundamentals	**Knowledge**: Understanding of different operating system architectures (e.g., monolithic, microkernel, hybrid). Familiarity with system calls, file systems, memory management, process scheduling, and input/output subsystems. **Skills**: Ability to configure and install various operating systems (e.g., Windows, Linux, macOS, UNIX). Proficiency in troubleshooting OS-related issues. **Attitude:** explain the core components of an operating system and how they interact, diagnose common OS-related performance issues, and optimize the OS for better resource utilization.
2	System Administration	**Knowledge**: Understanding of system administration tasks such as user management, system security, backup, software installation, and system configuration. **Skills**: Proficiency in managing user accounts, configuring system services, managing file systems, applying patches, and performing system backups. **Attitude**: perform routine administrative tasks, automate repetitive tasks through scripts, and ensure that the operating system is secure and running efficiently.
3	File System Management	**Knowledge**: Understanding of different types of file systems (e.g., NTFS, ext4, APFS, FAT32). Knowledge of file system structures, data storage, permissions, and disk management tools. **Skills**: Ability to create, mount, and manage file systems. Proficiency in handling disk partitions, optimizing file system performance, and resolving file system-related issues. **Attitude**: recover data from damaged or corrupted file systems, implement proper file access permissions, and ensure data integrity in an OS.
4	Memory Management	**Knowledge**: Understanding of memory hierarchy, paging, segmentation, and virtual memory. Knowledge of memory allocation techniques and OS strategies to handle memory (e.g., paging, swapping). **Skills**: Ability to monitor and troubleshoot memory usage in the operating system, such as detecting memory leaks, managing memory usage, and optimizing virtual memory. Attitude: design efficient memory management techniques for both hardware and software, ensure that system memory is allocated properly, and minimize the occurrence of out-of-memory errors.
5	Process Management	**Knowledge**: Understanding of how processes are created, scheduled, and terminated. Familiarity with multithreading, multitasking, process synchronization, and inter-process communication (IPC). **Skills**: Ability to manage processes using command-line tools or GUI utilities.

		Proficiency in using OS utilities to track process states and system resource consumption. **Attitude:** implement and manage process scheduling strategies, optimize resource allocation, and troubleshoot process deadlocks or race conditions.

3. Syllabus

OPERATING SYSTEMS
SEMESTER – III

Course Code	**M23BCS303**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours Theory + 20 Hours Practical**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Basic Understanding of Computer System Structure and Operating Systems Structure
- Analyze the main tasks carried out by the operating systems Process and thread management, CPU scheduling algorithms, memory management and deadlocks.
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.

Module -1

Introduction to operating systems: What operating systems do; Computer System organization; Computer System architecture; Operating System operations(dual-mode and multi mode);computing environments; **System structures:** Operating system services; User - Operating System interface; System calls; Types of system calls; operating system structures.

textbook 1- chapter 1(1.1,1.2,1.3,1.5,1.11) chapter 2 (2.1,2.2,2.3,2.4,2.7)

Module -2

Process management: Process Concept; Process Scheduling; Operation on Process; Inter-Process Communication. **Multithreaded programming**: Overview; Multicore programming, multithreading models, thread libraries, threading issues. **Process scheduling** - Basic Concepts, CPU I/O Burst Cycle; CPU Scheduler – Pre-emptive Scheduling, Dispatcher; Scheduling Criteria; Scheduling Algorithms – FCFS, SJF, Round-Robin, Priority.

textbook 1-chapter 3(3.1,3.2,3.3,3.4) chapter 4(4.1,4.2,4.3,4.4,4.6) chapter 6(6.1,6.2,6.3)

Module -3

Process Synchronization: Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Mutex locks; Semaphores; Classical problems of synchronization; **Deadlocks**: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

textbook 1-chapter 5(5.1,5.2,5.3,5.4,5.5,5.6,5.7) chapter 7(7.1-7.7)

Module -4

Memory Management: Background; Swapping; Contiguous memory allocation; Segmentation; Paging; Structure of page table; **Virtual Memory Management**: Background; Demand paging; Page replacement; Allocation of frames; Thrashing.

textbook 1-chapter 8(8.1-8.6) chapter 9(9.1,9.2,9.4,9.5,9.6)

Module -5

File System, Implementation of File System: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; **Implementing File system**: File system structure; File system implementation; Allocation methods; Free space management. **Storage management:** overview of Mass storage structures; Disk structure; Disk attachment; Disk scheduling;

textbook 1-chapter 11(11.1-11.5)chapter 12(12.1-12.5)chapter 10(10.1-10.4)

Description:

Implement all the program in c programming language.

PRACTICAL COMPONENT

1.	Develop a c program to implement the Process system calls (fork (), exec(), wait(), create process, terminate process)

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

2.	Simulate the following CPU scheduling algorithms to find turnaround time and waiting time a)FCFS b)Round Robin
3.	Develop a C program to simulate producer-consumer problem using semaphores
4.	Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.
5.	Develop a C program to simulate Bankers Algorithm for DeadLock Avoidance
6.	Develop a C program to simulate the following contiguous memory allocation Techniques: a) Worst fit b) Best fit
7.	Develop a C program to simulate page replacement algorithms: a) FIFO b) LRU
8.	Simulate following File Organization Techniques a) Single level directory b) Two level directory
9.	Develop a C program to simulate the Linked file allocation strategies
10.	Develop a C program to simulate SCAN disk scheduling algorithm

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts 9th edition, Wiley-India, 2018
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013

Reference Books:

1. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
2. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

Web links and Video Lectures (e-Resources):□

https://nptel.ac.in/courses/106105214□□

https://www.geeksforgeeks.org/operating-systems/

tps://www.youtube.com/playlist?list=PLBlnK6fEyqRiVhbXDGLXDk_OQAeuVcp2O

Practice exercises for the lab portion that are not on the syllabus:

1. Simulate the SJF CPU scheduling algorithm to find turnaround time and waiting time
2. Simulate the PRIORITY CPU scheduling algorithms to find turnaround time and waiting time
3. Develop a C program to simulate Dining philosophers problem using semaphores
4. Develop a C program to simulate Bounded-buffer problem using semaphores
5. Develop a C program to simulate Deadlock detection algorithm for Deadlock Detection
6. Develop a C program to simulate the FIRST FIT contiguous memory allocation Technique
7. Develop a C program to simulate page replacement algorithm OPTIMAL
8. Simulate following File Organization Techniques
   a) Single level directory b) Two level directory
9. Develop a C program to simulate the Indexed allocation strategies
10. Develop a C program to simulate LOOK disk scheduling algorithm

4.Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction to operating systems	**Competency**: **Understanding of Operating System Concepts** **Knowledge**: how operating systems manage hardware resources efficiently. **Skills**: Understanding of basic operating system functions and objectives, knowledge of operating system history and evolution.
2	Week 3-4: Process management	**Competency**: **Problem-solving Skills** **Knowledge**: develop skills in breaking down problems and designing efficient solutions. **Skills:** Proficiency in evaluating CPU scheduling algorithms, ability to analyze performance metrics.
3	Week 5-6: Process Synchronization and deadlocks	**Competency: Concurrency and Parallelism** **Knowledge**: to design and implement concurrent programs that utilize multiple threads or processes. **Skills**: Understanding process and thread concepts, synchronization mechanisms.
4	Week 7-8: Memory Management	**Competency: Memory Management** **Knowledge:** how operating systems manage memory resources efficiently to support multiple processes.

		Skills: Understanding of virtual memory concepts, familiarity with memory allocation strategies.
5	Week 9-10: File System, Implementation of File System and storage management	**Competency: File Systems** **Knowledge**: how operating systems manage storage devices and provide a unified interface for file management. **Skills**: Proficiency in file system organization and implementation, knowledge of disk management techniques.
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with programming assignments

5 Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of Verilog concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Theory Course with 4 credits: Integrated Professional Core Course (IPC)

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Theory (A)	Internal Assessment-Tests (A)	2	**60%**	**15**	**06**
	Assignments/Quiz/Activity (B)	2	**40%**	**10**	**04**
	TotalMarks		**100%**	**25**	**10**
Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Laboratory(B)	Record Writing	Continuous	**60%**	15	06
	Test at the end of the semester	1	**40%**	10	04
	TotalMarks		**100%**	**25**	**10**

FinalCIE Marks =(A) + (B)

Semester End Examination pattern:

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question

paper shall be English unless otherwise it is mentioned.

2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. The question paper may include at least one question from the laboratory component.
5. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Operating System Fundamentals**	Students should grasp the basic concepts, components, and functions of an operating system, including process management, memory management, file systems, and device management.
2	**Process Management**	Learning how processes are created, scheduled, and managed by the operating system, including topics such as process synchronization, inter-process communication, and deadlock handling.
3	**Concurrency and Synchronization**	Learning about concurrent processes, critical sections, mutual exclusion, synchronization primitives, and techniques for ensuring thread safety and avoiding race conditions.
4	**Memory Management**	Understanding memory hierarchy, virtual memory, memory allocation strategies, and techniques for efficient memory usage, including paging, segmentation, and memory protection.
4	**File Systems**	Exploring file system organization, file operations, directory structures, file system implementation, and techniques for improving file system performance and reliability.

8. Course Outcomes (Cos) and Mapping with POs/ PSOs
Course Outcomes (Cos)

Cos	Description
M23BCS303.1	Understand the fundamental concepts and principles of operating systems.
M23BCS303.2	Analyze various inter-process communication, multiprogramming mechanisms and apply different process scheduling algorithms.
M23BCS303.3	Examine multiple mechanisms for managing deadlock situations and Implement both software and hardware solutions to address the critical-section problem
M23BCS303.4	Implement and evaluate memory management techniques.
M23BCS303.5	Examine the structure of file systems and the organization of secondary storage devices

CO-PO-PSO Mapping

Cos/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS303.1	**3**	**-**	**-**	**-**		**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**	
M23BCS303.2	**-**	**-**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**	
M23BCS303.3	**-**	**-**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**3**		**3**
M23BCS303.4	**3**	**-**	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**	
M23BCS303.5	**-**	**-**	**3**	**-**		**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	
M23BCS303	**3**		**3**	**2**	**3**							**3**	**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**

Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks.

11. Future with this Subject

The future with operating systems is likely to involve several key trends and developments**:**

1. Continued Integration of AI: Operating systems will increasingly integrate AI and machine learning algorithms to provide more personalized and adaptive user experiences. This could involve features like predictive behavior, context-awareness, and intelligent automation.
2. Enhanced Security: With cyber threats evolving rapidly, future operating systems will place even greater emphasis on security. This might include built-in encryption, advanced authentication methods like biometrics, and more robust intrusion detection systems.
3. Interconnectivity and IoT: As the Internet of Things (IoT) expands, operating systems will need to seamlessly integrate with a wide range of devices and platforms. This could lead to more standardized communication protocols and frameworks for managing diverse IoT ecosystems.
4. Edge Computing: With the proliferation of edge computing devices, operating systems will need to support distributed computing architectures effectively. This involves optimizing resource management, latency reduction, and ensuring seamless connectivity between edge devices and centralized servers.

3rd Semester	**Professional Course (PC)** **DATA STRUCTURES AND ITS APPLICATIONS**	**M23BCS304**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming Skills**	A good grasp of programming fundamentals is essential. Students should be comfortable with at least one programming language, such as Python, Java, C++, or similar.
2	**Understanding of Algorithms**	Familiarity with basic algorithms and problem-solving techniques is crucial. This includes knowledge of concepts like sorting, searching, recursion, and complexity analysis.
3	**Mathematics**	While not always a strict requirement, a basic understanding of mathematical concepts such as sets, functions, and basic discrete mathematics can be helpful, especially for analyzing algorithm complexity.
4	**Data Representation**	Knowledge of different data types and their representations in memory, such as arrays, linked lists, stacks, queues, trees, and graphs
5	**Logic and Problem-Solving Skills**	Ability to think logically and analytically to solve complex problems efficiently
6	**Software Development Fundamentals**	Understanding of software development methodologies, version control systems (like Git), debugging techniques, and basic software engineering principles.
7	**Linear Algebra (Optional)**	Some advanced topics in data structures and algorithms, particularly those related to graph theory and optimization, may require a basic understanding of linear algebra concepts

2. Competencies

Sl. NO	Competency	KSA Description
1	**Understanding Basic Data Structures**	**Knowledge:** Knowledge of fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables, and heaps, including their properties and use cases. **Skills:** Ability to identify and explain different data structures, their theoretical concepts, and their advantages and limitations. **Attitudes:** Willingness to explore new data structures and understand their role in solving computational problems.
2	**Implementation of Data Structures**	**Knowledge:** Understanding of algorithms, operations, and memory management techniques for implementing data structures. **Skills:** Proficiency in coding and implementing data structures in programming languages, adhering to best practices. **Attitudes:** Attention to detail and a commitment to writing efficient and clean code.
3	**Applications of Data Structures**	**Knowledge:** Awareness of specific problems and scenarios where data structures can be optimally applied, such as using hash tables for fast retrieval or trees for hierarchical representation. **Skills:** Ability to match real-world problems to suitable data structures for efficient solutions. **Attitudes:** Curiosity to identify practical use cases and enthusiasm for experimenting with real-world problems.
4	**Sorting and Searching Algorithms**	**Knowledge:** Knowledge of common sorting (e.g., bubble sort, merge sort, quick sort) and searching algorithms (e.g., linear search, binary search), including their time and space complexities. **Skills:** Ability to implement and optimize sorting and searching algorithms for

		various datasets. **Attitudes:** Analytical mindset and persistence in improving algorithm performance.
5	**Graph Algorithm**	**Knowledge:** Understanding of graph traversal techniques such as depth-first search (DFS) and breadth-first search (BFS), and their applications. **Skills:** Skill in implementing graph algorithms and solving problems like pathfinding, network design, and social network analysis. **Attitudes:** Openness to learning advanced concepts and patience in solving complex problems involving graphs.
6	**Memory Management**	**Knowledge:** Knowledge of memory allocation, deallocation techniques, and dynamic memory management in programming. **Skills:** Ability to implement and manage dynamic data structures like linked lists, trees, and graphs while optimizing memory usage. **Attitudes:** Awareness of the importance of resource efficiency and a proactive approach to managing memory effectively.
7	**Practical Problem Solving**	**Knowledge:** Familiarity with diverse problem-solving techniques and paradigms for data structures and algorithms. **Skills:** Expertise in solving complex problems through hands-on coding assignments, projects, and competitive programming. **Attitudes:** Enthusiasm for challenging oneself with new problems and a growth-oriented mindset.
8	**Critical Thinking and Analysis**	**Knowledge:** Understanding of how to analyze problem requirements and assess the most suitable data structures and algorithms. **Skills:** Ability to evaluate multiple solutions and select the most efficient approach for implementation. **Attitudes:** Commitment to thorough analysis and a problem-solving attitude focused on efficiency and effectiveness.

3. Syllabus

Data Structures and its Applications **SEMESTER – III**			
Course Code	**M23BCS304**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Course objectives: This course will enable students to 1. To explain fundamentals of data structures and their applications. 2. To illustrate representation of Different data structures such as Stack, Queues, Linked Lists, Trees and Graphs. 3. To Design and Develop Solutions to problems using Linear Data Structures 4. To discuss applications of Nonlinear Data Structures in problem solving. 5. To introduce advanced Data structure concepts such as Hashing and Optimal Binary Search Trees			
Module -1			
INTRODUCTION TO DATA STRUCTURES: Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations Review of pointers and dynamic Memory Allocation. ARRAYS and STRUCTURES: Arrays, Dynamic Allocated Arrays, Structures and Unions, Polynomials, Sparse Matrices, representation of Multidimensional Arrays, Strings Text Book: Chapter-1:1.2 Chapter-2: 2.1 to 2.7 Reference Book 1: 1.1 to 1.4			
Module -2			

STACKS: Stacks, Stacks Using Dynamic Arrays, Evaluation and conversion of Expressions **QUEUES**: Queues, Circular Queues, Using Dynamic Arrays, Multiple Stacks and queues. **Text Book: Chapter-3: 3.1 to 3.4,3.6, 3.7**
Module -3
LINKED LISTS: Singly Linked, Lists and Chains, Representing Chains in C, Linked Stacks and Queues, Polynomials, Additional List Operations, Sparse Matrices, Doubly Linked List. **Text Book: Chapter-4: 4.1 to 4.5,4.7,4.8**
Module -4
TREES: Introduction, Binary Trees, Binary Tree Traversals, Threaded Binary tree, Binary Search trees, Selection Trees, Forests. **Text Book: Chapter-5: 5.1 to 5.3, 5.5, 5.7 to 5.9**
Module -5
GRAPHS: The Graph Abstract Data Types, Elementary Graph Operations. **HASHING:** Introduction, Static Hashing, Dynamic Hashing **EFFICIENT BINARY SEARCH TREES:** Optimal Binary Search Trees **Text Book:Chapter-6: 6.1, 6.2 Chapter 8: 8.1 to 8.3 Chapter 10: 10.1**
Text Books: 1.Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014 2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014. **Reference Books:** 1. Gilberg &Forouzan, Data Structures: A Pseudocode approach with C, 2nd Ed, Cengage Learning,2014. 2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012. 3. Jean-Paul Tremblay &Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Ed, McGraw Hill, 2013 4. A M Tenenbaum, Data Structures using C, PHI, 1989 5. Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week1-2: Introduction, Arrays, Strings**	**Data Structures, Classifications (Primitive &Non-Primitive), Data Structure Operations**: **Competence**: Understanding the fundamental concepts of data structures, their classifications, and basic operations. **Knowledge**: Knowledge of various data structures such as arrays, linked lists, stacks, queues, trees, graphs, etc. Understanding the differences between primitive (int, float, char, etc.) and non-primitive (arrays, structures, pointers, etc.) data types. **Skills**: Ability to implement and manipulate data structures efficiently, perform operations like insertion, deletion, traversal, and searching. **Review of Arrays, Structures, Self-Referential Structures, and Unions**: **Competence**:_Being familiar with the concepts and basic operations related to arrays, structures, self-referential structures, and unions. **Knowledge**:_Understanding how arrays store data linearly in memory, the concept of structures for organizing related data, self-referential structures like linked lists and trees, and unions for storing different data types in the same memory location. **Skills**:_Proficiency in working with arrays, structures, linked lists, trees, and unions. Ability to design and manipulate complex data structures efficiently. **Pointers and Dynamic Memory Allocation Functions**: **Competence**:_Understanding the concept of pointers and dynamic memory allocation. **Knowledge**:_Knowledge of how pointers store memory addresses and their use in accessing and manipulating data dynamically allocated memory. Understanding functions like malloc(), calloc(), realloc(), and free() for dynamic memory allocation

		and deallocation. **Skills**:_Proficiency in working with pointers, dynamic memory allocation, and memory management. Ability to avoid memory leaks and efficiently utilize memory resources. **Representation of Linear Arrays in Memory Dynamically Allocated Arrays, Array Operations**: **Competence**: Understanding how linear arrays are represented in memory and their dynamic allocation. **Knowledge**: Knowledge of how dynamically allocated arrays are created using pointers and dynamic memory allocation functions. Understanding various array operations such as insertion, deletion, searching, sorting, etc. **Skills**:_Ability to implement dynamic arrays, perform array operations efficiently, and manage memory dynamically allocated for arrays. **Strings: Basic Terminology, Storing, Operations, and Pattern Matching Algorithms**: **Competence**: Understanding the concept of strings and basic string operations. **Knowledge**: Knowledge of how strings are stored in memory, basic string manipulation operations like concatenation, comparison, substring extraction, etc., and pattern matching algorithms like brute force, KMP, Rabin-Karp, etc. **Skills**:__Proficiency in working with strings, implementing string operations efficiently, and applying pattern matching algorithms to solve string-related problems.
2	Week 3-4: **Stacks, Queues**	**Stacks**: **Competence**:_Understanding the concept of a stack, its properties, and basic operations. **Knowledge**: Knowledge of how a stack works, including its LIFO (Last In, First Out) property, and operations such as push (insertion), pop (deletion), peek (accessing the top element without removing it), and isEmpty (checking if the stack is empty). **Skills**: Proficiency in implementing stack operations efficiently, understanding array representation of stacks, and implementing stacks using dynamic arrays. Ability to apply stack-based algorithms for applications like Polish notation, infix-to-postfix conversion, and evaluation of postfix expressions. **Queues**: **Competence**: Understanding the concept of a queue, its properties, and basic operations. **Knowledge**: Knowledge of how a queue works, including its FIFO (First In, First Out) property, and operations such as enqueue (insertion), dequeue (removal), peek (accessing the front element without removing it), and isEmpty (checking if the queue is empty). **Skills**: Proficiency in implementing queue operations efficiently, understanding array representation of queues, and implementing circular queues and dynamic arrays for efficient memory management. Ability to apply queue-based algorithms for applications like solving the A Mazing Problem, implementing deques, and priority queues.
3	Week 5-6: **Linked Lists**	**Linked Lists**: **Competence**: Understanding the concept of linked lists, their properties, and various types. **Knowledge**: Knowledge of how linked lists are structured with nodes containing data and pointers to the next (and optionally, previous) nodes. Understanding memory allocation for linked list nodes and the concept of garbage collection for reclaiming memory occupied by unused nodes. Knowledge of different types of linked lists such as singly linked lists, doubly linked lists, and circular linked lists. **Skills**: Proficiency in implementing linked list operations like insertion, deletion,

		traversal, and searching efficiently. Ability to manage memory allocation and deallocation for linked list nodes, handle pointer manipulations, and implement different types of linked lists. Understanding the use of linked lists in implementing stacks, queues, and other data structures efficiently. **Applications of Linked Lists**: **Competence**: Understanding how linked lists can be applied to solve various problems. **Knowledge**: Knowledge of applications where linked lists offer advantages, such as representing polynomials and sparse matrices. Understanding how linked lists can efficiently handle sparse data structures by only storing non-zero elements and their indices. **Skills**: Ability to design and implement algorithms using linked lists for specific applications like polynomial manipulation and sparse matrix representation. Proficiency in optimizing algorithms to utilize the advantages offered by linked lists, such as efficient insertion and deletion operations.
4	Week 7-8: **Trees, Binary Tree Traversals**	**Trees**: **Competence**: Understanding the hierarchical data structure represented as a tree. **Knowledge**: Knowledge of the basic concepts of trees, including nodes, edges, root, parent, child, sibling, leaf, subtree, etc. Understanding the different types of trees, such as binary trees, binary search trees, and selection trees. Knowing the terminology associated with trees, including height, depth, level, and balanced trees. **Skills**: Proficiency in visualizing and interpreting tree structures, identifying and describing tree properties, and understanding tree-related algorithms and operations. **Binary Trees**: **Competence**: Understanding binary trees and their properties. **Knowledge**: Knowledge of binary trees, where each node has at most two children. Understanding the different types of binary trees, such as full binary trees, complete binary trees, perfect binary trees, etc. Knowing the various traversal methods for binary trees, including preorder, inorder, postorder, and level order traversals. **Skills**: Ability to implement binary tree data structures, perform binary tree traversals recursively or iteratively, and understand the applications of binary trees in data structures and algorithms. **Threaded Binary Trees**: **Competence**: Understanding threaded binary trees and their advantages. **Knowledge**: Knowledge of threaded binary trees, which are binary trees with additional pointers (threads) that make traversal operations more efficient. Understanding the different types of threaded binary trees, such as single-threaded and double-threaded binary trees. **Skills**: Proficiency in implementing threaded binary trees, understanding the threading process, and performing threaded tree traversals. **Binary Search Trees**: **Competence**: Understanding binary search trees (BSTs) and their properties. **Knowledge**: Knowledge of BSTs, which are binary trees with the property that for each node, all nodes in its left subtree have values less than the node's value, and all nodes in its right subtree have values greater than the node's value. Understanding BST operations, including insertion, deletion, searching, and traversal. **Skills**: Ability to implement BST data structures, perform BST operations efficiently, and understand the applications of BSTs in searching and sorting algorithms. **Selection Trees and Forests**: **Competence**: Understanding selection trees and forests. **Knowledge**: Knowledge of selection trees, which are specialized binary trees used for finding the kth smallest/largest element in a set of elements. Understanding the concept of forests, which are collections of disjoint trees. **Skills**: Proficiency in implementing selection trees and understanding their algorithms for finding the kth smallest/largest element efficiently. Ability to work

		with forests, including operations such as merging trees and finding connected components.
5	Week 9-10: **Graphs, Hashing**	**Graphs**: **Competence**: **Understanding**: The ability to understand the concept of graphs and their significance in representing relationships between objects. **Terminology**: Familiarity with graph terminologies such as vertices, edges, directed and undirected graphs, weighted and un-weighted graphs, cycles, paths, etc. **Knowledge**: **Representation**: Understanding different ways to represent graphs including adjacency matrix and adjacency list. **Operations**: Knowledge of elementary graph operations such as adding or removing vertices and edges, checking for connectivity, finding shortest paths, etc. **Skills: Implementation**: Proficiency in implementing graph data structures and associated algorithms for operations like traversal, shortest path finding, and cycle detection. **Analysis**: Ability to analyze graph algorithms in terms of time and space complexity, and understand their efficiency and limitations. **Hashing**: **Competence**: **Understanding**: Grasping the concept of hashing and its importance in fast data retrieval. **Organization**: Understanding different hash table organizations such as chaining, open addressing, and probing. **Knowledge**: **Functions**: Knowledge of hashing functions and their role in mapping keys to hash table slots efficiently. **Static and Dynamic Hashing**: Understanding the concepts of static and dynamic hashing, collision resolution techniques, and resizing strategies. **Skills**: **Implementation**: Proficiency in implementing hash tables using various organizations and hashing functions. **Optimization**: Ability to design efficient hashing functions and choose appropriate collision resolution strategies based on the application requirements. **Adaptation**: Skill in implementing dynamic hashing techniques to handle changes in the size of the dataset dynamically.
6	Week 11-12:	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with programming assignments

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with realworld competencies
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies

9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20 marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Fundamentals**	Students will gain a deep understanding of fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables. Understand their properties, operations, and use cases.
2	**Implementation Skills**	Students will develop proficiency in implementing data structures and associated algorithms in a programming language such as C, C++, Java, or Python.
3	**Problem-Solving Skills**	Enhance problem-solving skills by applying data structures and algorithms to solve a wide range of computational problems. This includes tasks like searching, sorting, traversing graphs, and optimizing resource usage.
4	**Efficiency and Optimization**:	Students will Learn techniques for optimizing the performance of algorithms and data structures, including strategies for reducing time complexity, space complexity, and improving overall efficiency.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS304.1	Explain different data structures and their applications.
M23BCS304.2	Apply Arrays, Stacks and Queue data structures to solve the given problems.
M23BCS304.3	Use the concept of linked list in problem solving.
M23BCS304.4	Develop solutions using trees and graphs to model the real-world problem.
M23BCS304.5	Explain the advanced Data Structures concepts such as Hashing Techniques and Optimal Binary Search Trees.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS304.1	3	-	-	-	-	-	-	-	-	-	-	-	2	-
M23BCS304.2	-	3	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS304.3	3	-	-	-	-	-	-	-	-	-	-	-	2	-

M23BCS304.4	**-**	**3**	**3**	-	-	-	-	-	-	-	-	-	**3**	-
M23BCS304.5	**3**	**-**	-	-	-	-	-	-	-	-	-	-	**-**	-
M23BCS304	**3**	**3**	**3**	-	-	-	-	-	-	-	-	-	**2.5**	-

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks.

10. Future with this Subject

The "Data Structure and its application" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The future with data structures is likely to see continued evolution and innovation, driven by the growing complexity and scale of data processing tasks in various domains such as artificial intelligence, big data analytics, and distributed systems. Here are some potential directions for the future of data structures.

- **Efficiency Improvements:** There will be a continued focus on developing data structures that offer better time and space complexity characteristics to handle increasingly large datasets and computational tasks. This might involve refining existing data structures or inventing new ones optimized for specific use cases.
- **Adaptation to New Technologies:** As new computing paradigms like quantum computing and neuromorphic computing emerge, data structures will need to be adapted or even completely reimagined to take advantage of these technologies' unique capabilities and constraints.
- **Data Structure Fusion:** There might be efforts to combine multiple data structures into unified structures that offer the benefits of each component while minimizing their individual drawbacks. This could lead to more efficient and versatile data handling techniques.
- **Dynamic Data Structures:** With the rise of dynamic and streaming data sources, there will be a greater need for data structures that can efficiently handle data arriving in real-time or in a continuously changing manner. Dynamic data structures that can adapt their shape and organization dynamically will be crucial.
- **Parallel and Concurrent Data Structures:** As parallel and distributed computing become increasingly prevalent, there will be a demand for data structures that can effectively handle concurrent access from multiple threads or nodes without sacrificing performance or data integrity.
- **Data Structures for AI and Machine Learning:** With the growing importance of artificial intelligence and machine learning, there will be a need for specialized data structures optimized for

tasks such as representing neural networks, handling high-dimensional data, and efficiently storing and querying large training datasets.

- **Privacy-Preserving Data Structures:** As concerns about data privacy and security continue to rise, there will be interest in developing data structures that enable secure computation and querying while protecting sensitive information.
- **Environmental Considerations:** Given the increasing focus on sustainability and energy efficiency in computing, there might be efforts to design data structures that minimize energy consumption and environmental impact, for example, by reducing memory usage or optimizing cache performance. Overall, the future of data structures is likely to be characterized by ongoing innovation driven by the evolving needs of various applications and the opportunities presented by advances in technology.

3rd Semester	**Professional Course (PC)** **SOFTWARE ENGINEERING**	**M23BCS305**

1. Prerequisites

S/L	Proficiency	Prerequisites
01	Understanding of Basic Programming Concepts	Solid understanding of basic programming concepts and data structures. This includes knowledge of variables, control structures (loops, conditionals), arrays, linked lists, stacks, queues, trees, and graphs.
02	Mathematical Foundations	A strong foundation in mathematics is essential. This includes knowledge of algebra, calculus, and discrete mathematics. Understanding of topics such as functions, equations, inequalities, sequences, series, and combinatory is crucial for analyzing algorithmic complexity.
03	Programming Proficiency	Proficiency in at least one programming language is necessary. Understand concepts such as loops, recursion, functions, and basic algorithmic constructs.
04	**Problem Solving Skills**	Possess strong problem-solving skills. This involves the ability to analyze problems, identify patterns, and devise algorithmic solutions. Prior experience with problem-solving competitions or algorithmic challenges can be beneficial.
05	**Logical Thinking**	The ability to think logically and critically is crucial for understanding complex algorithms and analyzing their performance.

2. Competencies

S/L	Competency	KSA Description
1.	Understanding of Software Engineering Principles and Practices	**Knowledge:** Knowledge of Software methodologies such as Life cycle, Coding and testing. **Skills:** Understanding of Software principles and values **Attitudes:** Appreciation for the Familiarity with frameworks and their application in software development.
2.	Knowledge of testing and Coding Concepts and Practices	**Knowledge:** Understanding of coding principles, including collaboration, automation, measurement, and sharing (CAMS). **Skills:** Knowledge of Software practices such as continuous integration, continuous delivery, infrastructure as code, and automated testing. **Attitudes:** Appreciation for the awareness of Software tools and technologies used for deployment, monitoring, and orchestration.
3.	Technical Knowledge	**Knowledge:** Proficiency in programming languages commonly used in software development (e.g., Java, Python, JavaScript). **Skills:** Understanding of version control systems (e.g.,Git) and their role in collaborative development. **Attitudes:** Valuing the importance of Knowledge of containerization technologies and container platforms.

3. Syllabus

Software Engineering **SEMESTER – III**			
Course Code	**M23BCS305**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- To provide the idea of decomposing the given problem into Analysis, Design, Implementation, Testing and Maintenance phases.
- To provide an idea of using various process models in the software industry according to given circumstances.

- To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Module -1

Introduction: FAQ's about Software Engineering, Professional and Ethical Responsibility.

Socio-Technical systems: Emergent system properties; Systems Engineering; Organizations, people and computer systems; Legacy systems.

Ch -1 Ch-2

Module -2

Critical Systems: A simple safety- critical system; System dependability; Availability and reliability, safety, security.

Software Processes Models, Process iteration, Process activities, The Rational Unified Process, Computer Aided Software Engineering.

Ch -3 Ch-4

Module -3

Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document.

Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management.

Ch -6 Ch -7

Module -4

System Models: Context models; Behavioral models; Data models; Object models; Structured methods.

Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles, reference architectures

Application architectures: Data-processing systems, Transaction-processing systems, Event- processing systems Language-processing system

Ch-8 Ch-11 Ch-13

Module -5

Rapid software development: Agile methods, Extreme programming, Rapid application development, Software prototyping.

Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods.

Software testing: System testing; Component testing; Test case design; Test automation.

Ch- 17 Ch-22 Ch-23

Textbooks

1. Ian Somerville: Software Engineering, 8 th Edition, Pearson Education.
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman, McGraw-Hill International edition.

Reference Textbooks

1. An Integrated Approach to Software Engineering, by Pankaj Jalote, Narosa Publishing House.
2. James F Peters and WitoldPedryez, "Software Engineering – An Engineering Approach", John Wiley and Sons, New Delhi, 2000.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	Introduction: FAQ's about Software Engineering, Professional and Ethical Responsibility. Socio-Technical systems: Emergent system properties; Systems Engineering; Organizations, people and computer systems; Legacy systems.
2	Week 3-4:	Critical Systems: A simple safety- critical system; System dependability; Availability and reliability, safety, security. Software Processes Models, Process iteration, Process activities, The Rational Unified Process, Computer Aided Software Engineering.
3	Week 5-6:	Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document. Requirements Engineering Processes: Feasibility studies; Requirement's elicitation and analysis; Requirement's validation; Requirements

		management.
4	Week 7-8:	System Models: Context models; Behavioral models; Data models; Object models; Structured methods. Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles, reference architectures Application architectures: Data-processing systems, Transaction-processing systems, Event- processing systems Language-processing system
5	Week 9-10:	Rapid software development: Agile methods, Extreme programming, Rapid application development, Software prototyping. Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods. Software testing: System testing; Component testing; Test case design; Test automation.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application
4	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
5	Multiple Representations	Introduce topics in various representations to reinforce competencies
6	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies
7	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject (duration 03 hours)

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding Software Engineering Fundamentals	Students will comprehend the definition and significance of Software Engineering in problem-solving.

2	Analyzing Software Efficiency	Students will apply a structured framework to assess software performance in terms of time and space complexity.
3	Mastering in Planning Approaches	Students will gain proficiency in implementing and evaluating Software by Good Planning.
4	Exploring Various Testing Strategies	Students will explore and apply strategies like Black and White Box testing.
5	Understanding Software Limitations and Coding Strategies	Students will identify limitations of Software's including completeness, and apply coping strategies.
6	Synthesizing Knowledge and Applying in Problem Domains	Students will synthesize their understanding of various algorithmic techniques to tackle complex problems in real-world scenarios, demonstrating proficiency in algorithm design and analysis.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS305.1	Understand and apply the principles of software engineering and how they contribute to the development of software systems.
M23BCS305.2	Apply various software process models develop software projects according to project requirements.
M23BCS305.3	Analyze software requirements to identify potential risks, dependencies, and inconsistencies.
M23BCS305.4	Design software systems by integrating multiple engineering practices to solve complex problems.
M23BCS305.5	Assess software testing strategies to ensure the effectiveness and completeness of tests in verifying software functionality.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS305.1	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS305.2	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS305.3	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCS305.4	-	-	3-	-	-	-	-	-	-	-	-	-	3	3
M23BCS305.5	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCS305	-	3	3	-	-	-	-	-	-	-	-	-	3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	Total
Module 1	10					10
Module 2		10				10
Module 3			10			10
Module 4				10		10
Module 5					10	10
Total	10	10	10	10	10	50

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	Total
Module 1	20					20
Module 2		20				20
Module 3			20			20
Module 4				20		20
Module 5					20	20

Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting: Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Software Engineering" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

1. **Software Development and Engineering**
   Students can become software developers, systems architects, or software engineers, roles that are in high demand across industries like technology, finance, healthcare, and more. Companies such as Google, Microsoft, and Apple highly value these skills.
2. **Data Science and Big Data Analytics**
   Careers as data scientists, data analysts, and big data engineers are promising. These professionals are sought after in sectors such as e-commerce, healthcare, finance, and social media, where data-driven decision-making is pivotal.
3. **Artificial Intelligence (AI) and Machine Learning (ML)**
   Opportunities include roles as AI/ML engineers, data scientists, and research scientists. This field is rapidly growing and is fundamental to advancements in automation, robotics, and predictive analytics.
4. **Cyber Security**
   Students can pursue careers as cyber security analysts, cryptographers, or information security managers. These roles are critical in protecting digital assets and information systems from threats.
5. **Academic Research and Teaching**
   Students can become researchers or professors, contributing to academia by teaching and exploring new areas in algorithm design and analysis. This path is essential for the on-going development of computer science as a discipline.

3rd Semester	**Engineering Science Course (ES)** **UNIX AND SHELL PROGRAMMING**	**M23BCS306A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Computer Skills	Familiarity with using a computer, navigating the file system, managing files and directories, and using the command line interface (CLI) is essential.
2.	Understanding of Operating Systems	Basic knowledge of how operating systems work, including processes, memory management, file systems, and user permissions.
3.	Familiarity with Command Line Interface (CLI)	Understanding of how to navigate directories, list files, create directories, copy/move files, and execute commands using the command line interface.
4.	Programming Fundamentals	Basic understanding of programming concepts like variables, data types, loops, conditionals, functions, and control structures. This will help in understanding shell scripting.
5.	Text Editing Skills	Proficiency in using a text editor, as shell scripts are essentially text files containing commands.
6.	Problem-Solving Skills	Ability to analyze problems, break them down into smaller components, and devise solutions. Shell scripting often involves solving various problems efficiently.

2. Competencies

S/L	Competency	KSA Description
1.	**Proficiency in Command Line Interface**	**Knowledge:** Understand the fundamental of Command line Interface **Skills:** Efficient file manipulation, text processing, and system administrations. **Attitudes:** Be comfortable with command line interface
2.	**Shell Scripting**	**Knowledge:** Understanding shell script writing. **Skills:** Writing shell scripts to automate tasks, create custom utilities, and streamline workflows, **Attitudes:** Confident in writing shell scripts.
3.	**System Administration Skills**	**Knowledge:** Understanding system administration. **Skills:** User management, file permissions, process management, and system monitoring. **Attitudes:** Confident in managing UNIX/Linux-based systems
4.	**Text Processing and Manipulation**	**Knowledge:** Understanding basic text processing and Manipulation. **Skills:** Manipulate and process text using command line tools and shell scripting, including tasks such as searching, filtering, sorting, and transforming text data. **Attitudes:** Comfortable in managing text.

3. Syllabus

UNIX and Shell Programming
SEMESTER – III

Course Code	M23BCS306A	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course Learning objectives:

The main objectives of this course are to:

- Understand the features, architecture of UNIX and its commands.
- Discuss different UNIX files, attributes and permissions.
- Discuss filter programs and regular expressions.
- Familiarize with advanced filters

Module -1

UNIX Architecture and Command Usage: UNIX Architecture, Features of UNIX, Internal and External Commands.**General-Purpose Utilities**: cal, date, echo, printf, bc, passwd, who, uname, tty, stty. **The File System**: The Parent-Child Relationship, the HOME variable, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames, ls: Listing Directory Contents. (Text book: 1 Chapters: 2.1, 2.2, 2.5, 3.1 to 3.5, 3.9 to 3.13, 4.1 to 4.11)
Module -2
Handling Ordinary Files: cat, cp, rm, mv, more, file, wc, od, cmp, comm, diff, Basic File Attributes: ls –l, the –d option, file ownership, file permissions, chmod, directory permissions, changing file ownership.**More File Attributes**: File Systems and Inodes, Hard Links, Symbolic Links and ln, umask Modification and Access Times. (Text book: 1 Chapters: 5.1 to 5.5, 5.7 to 5.12, 6.1 to 6.7, 11.1 to 11.6)
Module -3
Simple Filters: The sample database, paginating files, head, tail, cut, paste, sort, uniq, tr.**Filters using Regular Expression**: grep: Searching for a pattern, Basic Regular Expression (BRE), egrep: Extended Regular Expression. (Text book: 1 Chapters: 12.1, 12.9 to 12.9, 13.1 to 13.3)
Module -4
Essential Shell Programming Part I: Shell Scripts, read, Using command line arguments, exit and exit status of command, the logical operators && and || - conditional execution, the if conditional, using test and [ ] to evaluate expressions, the case conditional, expr, $0, while, for, set and shift, trap: Interrupting a program (Text book: 1 Chapters: 14.1 to 14.13)
Module -5
awk: An Advanced Filter:Simple awk filtering, Splitting a Line into Fields, printf: Formatting Output, Variables and Expressions, Comparison Operators, Number Processing, Built-in Variable, Arrays, Functions, Control Flow, Looping with for and while (Text book: 1 Chapters: 18.1 to 18.15).
TextBook(s)
1. UNIX – Concepts and Applications, Sumitabha Das, 4th Edition, McGraw Hill, 2017. 2. UNIX and SHELL Programming, Behrouz A Forouzan and Richard F Gilberg, India. Edition, Cengage Learning, Third Reprint 2008
ReferenceBooks
1. UNIX – The Complete Reference, Kenneth Rosen et al, 2nd Edition, Tata McGraw Hill Fourth Reprint 2008 2. Your UNIX: The Ultimate Guide, Sumitabha Das, McGraw Hill, 2001. 3. Introduction to UNIX and Shell Programming, M G Venkateshmurthy, Pearson Edition.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: UNIX Architecture and Command Usage	**Understand** UNIX Operating System Architecture. **Acquire the Knowledge**of UNIX command and its usage. Understand the file system.
2	Week 3-4: Handling Ordinary Files	**Impart the knowledge of** Command Line Interface. Different commands for handling files. Able to write shell scripts for handing files.
3	Week 5-6: Simple Filters and Filters using Regular Expression	Understand and apply simple Filters and Regular Expressions for solving various problems. Develop scripts for handling regular expressions
4	Week 7-8: Essential Shell Programming	**Acquire the Knowledge:** UNIX data types, operators, if conditional Statement and looping statements, etc. Use various UNIX features for developing scripts.
5	Week 9-10: awk- An Advanced Filters	**Understand the importance of** Advanced filters. Develop shell scripts using advanced filters.
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with programming assignments.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Live Demonstration	Develop and run Shell scripts in the classroom.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Programming Assignments	Assign programming tasks to improve the practical skills.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the architecture of Linux operating System	Students will understand the Linux operating system Architecture.
2	Analyze the working of various Linus Commands	Students will be able to analyze the working of various Linux commands by executing commands.

3	Develop a Shell Script	TocreateprogramsintheLinuxenvironmentusingLinuxutilities andcommands.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS306A.1	Understand the fundamental concepts of UNIX Operating system. Analyze working of various commands.
M23BCS306A.2	Apply various filters to solve variety of applications.
M23BCS306A.3	Develop Regular expressions for pattern matching.
M23BCS306A.4	Develop various shell scripts for performing various operations on Linux Operating System and use awk advanced filters.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS306A.1	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**2**
M23BCS306A.2	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**-**
M23BCS306A.3	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**1**	**-**
M23BCS306A.4	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**		**2**
M23BCS306A	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**1.5**	**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2		**10**			**10**
Module 3			**10**		**10**
Module 4				**10**	**10**
Module 5	**2**	**3**	**2**	**3**	**10**
Total	**12**	**13**	**12**	**13**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2		**20**			**20**
Module 3			**20**		**20**
Module 4				**20**	**20**
Module 5	**5**	**5**	**5**	**5**	**20**
Total	**25**	**25**	**25**	**25**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "UNIX and Shell Programming" course in the third semester of the B.E (Computer Science & Engineering Branches) program placesanimportant role for learning several future courses in the undergraduate program. This subject is very important for conducting many laboratory subjects such as Analysis and Design of Algorithm, Database Management System, Data Structures, Python programming, etc.

Here are some notable contributions:

- **Internet of Things (IoT) and Embedded Systems:** Many IoT devices and embedded systems run on UNIX-like operating systems or utilize shell scripts for managing system tasks. Understanding UNIX and shell programming is beneficial for developers working on IoT devices, embedded systems, and firmware development.

- **Education and Training:** UNIX and shell programming concepts are often taught in computer science and information technology curricula as foundational skills. Aspiring software engineers, system administrators, and IT professionals continue to learn UNIX and shell programming to build a strong technical foundation.
- **Data Processing and Analysis:** UNIX tools and shell scripting are commonly used for data processing, manipulation, and analysis tasks. As data continues to grow in volume and complexity, the ability to efficiently process and analyze data using command line tools and shell scripts remains relevant for data scientists, analysts, and researchers.

3rd Semester	**Engineering Science Course (ES)** **OOPs with JAVA**	**M23BCS306B**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Computer Skills	Basic computer skills, such as saving files in multiple versions and formats.
2	Programming Fundamentals	Familiar with general coding concepts like variables, basic data types, Conditional Statements, Looping, Functions, creation of source file, compilation process, program execution techniques.
3	Multi-Process Execution Programming	Familiar with the way to execute multiple tasks simultaneously by creating multiple processes.
4	Basic Object Orientation Concepts	Basic of four basic principles: encapsulation, inheritance, polymorphism, and abstraction. Where these four OOP principles can be used enable to create objects and collaborate to create powerful applications too.
5	Programming basic tools	Familiar with Programming tools like assemblers, compilers, linkers translate, flowchart, algorithms which can be used to form a program from a human write-able and readable source language into the bits and bytes that can be executed by a computer.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Object Oriented Concepts**	**Knowledge:** Importance of Object Orientation Concepts. Understanding of the basics of Object Orientation Programming. **Skills:** Ability to apply Object Orientation Concepts to create objects using appropriate structure. **Attitudes:** Appreciation to understand the importance of object orientation perspective and implement the same at basic level.
2	**Basic of Programming**	**Knowledge:** Understanding of basic elements of programming specific to Java Language. Basics of Java program execution. **Skills:** Designing basic Java program using basic elements of programming language. Creating and executing simple Java programs. **Attitudes:** Appreciation for the role of Java programming elements and its execution.
3	**Java Classes and its methods**	**Knowledge:** Understanding how classes are defined with data members and methods. **Skills:** Designing of classes for real world objects. Defining appropriate attributes and methods for classes. **Attitudes:** Valuing the importance of classes and its methods in line with real-world objects.
4	**Reusability of Classes and Methods**	**Knowledge:** Understanding the importance of code reusability through classes and methods reusability. **Skills:** Applying concepts of object orientation with classes and methods. Describing the actually importance of reusability through implementations. **Attitudes:** Openness to learning and using object orientation concepts to achieve code reusability.
5	**Exceptions and Handling the Exceptions**	**Knowledge:** Understanding of issues with exceptions. **Skills:** Implementing how to handle the exceptions through appropriate Java programming construct. **Attitudes:** Appreciation for the way exception is handled and making the execution of program in control.
6	**Multi-**	**Knowledge:** Understanding the characteristics and importance of parallel

	Threaded Programming	execution of a task. **Skills:** Designing and analyzing the parallel execution using thread concepts. Implementing multi-thread concepts. **Attitudes:** Recognizing the significance of flip-flops in sequential logic circuits

3. Syllabus

OOPs with JAVA
SEMESTER – III

Course Code	**M23BCS306B**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- To learn primitive constructs JAVA programming language.
- To understand Object Oriented Programming Features of JAVA.
- To gain knowledge on: packages, multithreaded programing and exceptions.

Module -1

Introduction to Java: Java Buzz Words, Java's Magic: Bytecode, Two Paradigms of OOPS, Principles of OOPS, Lexical Issues, Use Block of Code, First Java Program.

Variables in Java: Data Types, Literals Constants, Variables, Type Conversion.

Arrays: One-dimensional array, multi-dimensional array, alternative array declaration, Java Strings.

Operators: Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, The AssignmentOperator, The Conditional Operator, Operator Precedence.

Selected Topics from Chapter 1, 2, 3, 4

Module -2

Control Statements: Selection Statements, Iteration Statements, Jump Statements.

Introduction to Classes: Class Fundamentals, Declaring Objects, Object Reference Variables, Garbage Collection.

Methods: Introducing Class Methods, Constructors, This Keyword, Example Java Programs with Class and Methods.

Selected Topics from Chapter 5, 6, 7

Module -3

Polymorphism: Overloading Methods, Overloading Constructors, Using Objects as Parameters and Argument Passing, Returning Objects.

Access Control: Introduction to Access specifier, Understanding Static Members & Methods, Nested andInner Classes.

Inheritance: Inheritance Basics,Member access and inheritance,using super keyword, Constructors call in Multilevel Hierarchy, Method Overriding, Abstract Classes, Final and Inheritance.

Selected Topics from ***Chapter 7, 8***

Module -4

Packages: Packages, defining a Package, Access protection, Importing Packages.

Interfaces:Defining anInterfaces, implementing an interface, Access Implementations through interface references, Nested Interface.

Exceptions-Handling: Fundamentals of Exception-Handling Fundamentals, Exception Types, Using different types of Exception clauses, Nested try statements, Creating Your Own Exception Subclasses.

Selected Topics from ***Chapter 9, 10***

Module -5

Multithreaded Programming: The Java Thread Model, The Main Thread, creating a user thread, CreatingMultiple Threads, Synchronization, Inter-threadCommunication, Deadlocks, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State.

Selected Topics from ***Chapter 11***

Text Books:

1. Java: The Complete Reference, 8thEdition, by Herbert Schildt, November 2012, McGraw-Hill Edition 2011, ISBN: 978-1-25-900246-5.

2. Programming with Java A Primer, 4th Edition, by E Balagurusamy, Mar-2010, Tata McGraw Hill Education, ISBN: 978-0-07-014169-8.

Reference Books:

1. Programming with JAVA, 5th Edition, by M P Bhave and S A Patekar, 2017, Pushp Print Services, ISBN:978-81-317-2080-6.

2.Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006 (https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf)

Tutorial Components

1. Write and Execute a Java program to show how the different ways of declaring and initialization a Two-Dimensional array in Java.
2. Write and Execute a Java program to print list of student names using for-each loop.
3. Develop a class called Student with the data members USN, Name, IA1_Marks, IA2_Marks, IA_2 Marks and Avg_Marks and method ComputeAvg(m1,m2,m3) to compute the average of IA Marks. Develop the suitable class and main method for demonstration.
4. Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager, Developer, and Programmer. Each subclass should have properties such as name, address, salary, and job title. Implement methods for displaying salary with 10% raise for Programmer, 25% raise for Developer and 40% raise for Manager.
5. Write and Execute a Java program to show the order of constructor call and its execution in multi-level inheritance.
6. Write a Java program to create an interface Sortable with a method Sort() that sorts an array of integers in ascending order. Create two classes BubbleSort and SelectionSort that implement the Sortable interface and provide their own implementations of the Sort() method.
7. Demonstrate how MyPack package is created in Java with class called MyClass and method called MyMethod() and import the package MyPack in the file called New.java to declare object for the class MyClass and call the method MyMethod() in the main method of New.java file.
8. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowels.
9. Create a child thread by implementing the Runnable interface wherein the child thread does string concatenation, and the main thread changes the string to uppercase.
10. Write a Java program to Create three classes Storage, Counter and Printer. The Storage class should store an integer, the Counter class should create a thread that starts counting from 0 (like, 0,1,2,3,…) and stores each value in the Storage class. The Printer class should create thread that keeps reading the value from the Storage class and prints it.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction to Java, Variables in Java&Arrays	Basic Java Programming, Java Programming basic constructs and applying basic programming constructs in Java execution environment.
2	Week 3-4: Operators, Control Statements&Introduction to Classes	Java Looping Constructs and Java Classes, Basics of Java Classes with looping constructs and Designing and Implementing Classes in Java and Looping constructs.
3	Week 5-6: Methods, Polymorphism&Access Control	Class Methods with Polymorphism and Access Control, using methods in Java Classes and accessing the members and class using appropriate access control with polymorphism and designing and implementing class methods through polymorphism and access mechanism.
4	Week 7-8: Inheritance, Packages&Interfaces	Inheritance with Packages and Interfaces, importance of Inheritance, Use of Packages and Interfaces and Applying the concept of Inheritance with Classes, creating package and importing the same with interfaces.
5	Week 9-10: Exceptions-Handling& Introduction to Multithreaded Programming	Exceptions, Exception-handling & Introduction to multi-threading concepts, Understanding Exception, handling exceptions and basics of multi-threading and Implementing exception handlers by creating threads.
6	Week 11-12:	Understanding multi-threaded concepts with synchronization and inter-

	Multithreaded Programming	thread communications and Implementing and managing multi-threaded concepts through synchronization and inter-thread communications.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Image/Video/Animation	Incorporate visual aids like image/videos/animations to enhance understanding of programming constructs.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Programming-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding basic Java Programming Constructs	Students will grasp the fundamental concepts of Java Programming, including basic constructs.
2	Designing simple basic Java Programs	Students will learn to design and implement basic and simple Java programs.
3	Proficiency in Java Specific Constructs	Students will become proficient in understanding and applying the Java specific constructs to improve the efficiency of Java programming logics.

4	Programming-Based Learning	Through program execution-based learning, students will undergo the demonstration of Java programming constructs working principles.
5	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with Java Programming, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS306B.1	Understand and apply the basic programming constructs.
M23BCS306B.2	Apply the structure of classes and methods in Java programming environment.
M23BCS306B.3	Analyze the different programming constructs of Java and its effectiveness in improving the efficiency of Java programs.
M23BCS306B.4	Implement appropriate Java programming constructs to solve real-world problem sample scenarios.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS306B.1	**3**													
M23BCS306B.2	**3**												**3**	
M23BCS306B.3		**3**											**3**	**3**
M23BCS306B.4			**3**											**3**
M23BCS306B	**3**	**3**	**3**										**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2		**10**			**10**
Module 3			**10**		**10**
Module 4				**10**	**10**
Module 5	**2**	**3**	**2**	**3**	**10**
Total	**12**	**13**	**12**	**13**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**20**				**20**
Module 2		**20**			**20**
Module 3			**20**		**20**
Module 4				**20**	**20**
Module 5	**5**	**5**	**5**	**5**	**20**
Total	**25**	**25**	**25**	**25**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks.

10. Future with this Subject

The "OOPs with Java" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

- **Advanced Java:** The knowledge gained in this course, is very useful for learning advanced Java Programming.
- **Mobile Application Development:** Nowadays, mobile application developments are most popular. So, Java programming is very useful in mobile application development.

3rd Semester	Engineering Science Course (ES) OOPs with C++	M23BCS306C

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Basic Computer Skills**	Familiarity of different Operating Systems and the basic knowledge of command line usage are very needful.
2.	**Knowledge of Integrated Development Environment**	Requires the basic skills to use various tools like text editor, compiler, linker and C++ IDE.
3.	**Problem Solving Skills**	Knowledge of the Algorithmic thinking and Logical thinking needed.
4.	**Mathematics**	Proficiency in Mathematics required to find the roots of quadratic equation, Trigonometric Functions etc.
5.	**Basics of C Programming**	Fundamental understanding of C is essential for object-oriented programming. This includes syntax, data types, variables, control structures, functions, and pointers
6.	**Previous Coursework**	Completion of introductory courses in principles of programming in C related field.
7.	**Basic Computer Skills**	Familiarity of different Operating Systems and the basic knowledge of command line usage is very needful.
8.	**Knowledge of Integrated Development Environment**	Requires the basic skills to use various tools like text editor, compiler, linker and C++ IDE.
9.	**Problem Solving Skills**	Knowledge of the Algorithmic thinking and Logical thinking needed.
10.	**Mathematics**	Proficiency in Mathematics required to find the roots of quadratic equation, Trigonometric Functions etc.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction to Object Oriented Concepts**	**Knowledge:** Importance of Object Orientation Concepts. Understanding of the basics of Object Orientation Programming. **Skills:** Ability to apply Object Orientation Concepts to create objects using appropriate structure. **Attitudes:** Appreciation to understand the importance of object orientation perspective and implement the same at basic level.
2	**Basic of Programming**	**Knowledge:** Understanding of basic elements of programming specific to C++ Language. Basics of C++ program execution. **Skills:** Designing basic C++ program using basic elements of programming language. Creating and executing simple C++ programs. **Attitudes:** Appreciation for the role of C++ programming elements and its execution.
3	**C++ Classes and its methods**	**Knowledge:** Understanding how classes are defined with data members and methods. **Skills:** Designing of classes for real world objects. Defining appropriate attributes and methods for classes. **Attitudes:** Valuing the importance of classes and its methods in line with real-world objects.

4	**Reusability of Classes and Methods**	**Knowledge:** Understanding the importance of code reusability through classes and methods reusability. **Skills:** Applying concepts of object orientation with classes and methods. Describing the actually importance of reusability through implementations. **Attitudes:** Openness to learning and using object orientation concepts to achieve code reusability.
5	**Exceptions and Handling the Exceptions**	**Knowledge:** Understanding of issues with exceptions. **Skills:** Implementing how to handle the exceptions through appropriate C++ programming construct. **Attitudes:** Appreciation for the way exception is handled and making the execution of program in control.

3. Syllabus

OOPs WITH C++ **SEMESTER – III**			
Course Code	**M23BCS306C**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives:

- The objectives of the course are to have students identify and practice the object-orientedprogramming concepts and techniques, practice the use of C++ classes and class libraries, arrays, vectors, inheritance and file I/O stream concepts.

Module -1

Introduction to object Oriented Programming: OOP Paradigm, Basic concepts of OOP, Beginning with C++, Applications of C++, A simple C++ programs, Structure of C++ Program.

Tokens: Character sets and Symbols, Keywords, C++ Identifiers, Variables and Constants, Dynamic Initialization of variables, Reference variables, Basic data types in C++, User defined data types, Storage classes, Operators, Type cast Operators**.**

Module -2

Decision and Control Structures: if statement, if-else statement, switch statement, Loop: while, do while, for, Jump Statements: break, return, go to.

Classes and Objects: Classes in C, class declaration, declaring objects, Define member functions, call by reference, return by reference, inline functions, default arguments, Function Overloading.

Module -3

Constructor and Destructors: Constructors, Parameterized constructors, Multiple Constructors in a class, Constructors with default arguments, Dynamic initialization of Objects, Const object, Destructors.

Operator Overloading: Introduction, defining operator overloading, Overloading unary and binary operators, Type Conversions.

Module -4

Inheritance: Defining Derived classes, Types of Inheritance- Single inheritance, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid Inheritance, Abstract classes, constructors in derived class, Member classes. **Polymorphism:** Introduction, Virtual functions, virtual constructor and destructors.

Module -5

Exception Handling: Basic of Exception Handling, Exception Handling Mechanism, Throwing Mechanism, Catching Mechanism, Rethrowing an Exception, Exception in Operator overloaded functions.

The C++ I/O System Basics: C++ Streams, The C++ Classes, Formatted I/O File I/O: and File Classes, Opening and Closing a File, Reading and Writing Text Files, Detecting EOF.

List of Programs for Practice	
1	Design a C++ program to perform simple calculator.
2	An election is contested by five candidates. The candidates are numbered 1 to 5 and a voting is done by marking the candidate number in a ballot paper. Write a C++ program to read the ballot and count the votes cast for each candidate using an array variable count. In case, a number read is outside the range 1 to 5 the ballot should be considered as a 'spoilt ballot', and the program should also count the number of spoilt ballots.

3	Develop a C++ program to sort the elements in ascending and descending order
4	Develop a C++ program to demonstrate function overloading for the following prototypes. add(int a, int b) add(double a, double b)
5	Develop a C++ program using Operator Overloading for overloading Unary minus operator.
6	Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers.
7	Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.
8	Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.
9	Develop a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.
10	Develop a C++ program that handles array out of bounds exception using C++.

Text Books:

1. Balagurusamy E, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd., Sixth Edition 2016.
2. Herbert schildt, The Complete Reference C++, 4th edition, TMH, 2005

Reference Books:

1. D.S Guru, Object- Oriented Programming with C++.
2. C++ and Object Oriented Programming – Jana, PHI Learning.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2:** Introduction to object Oriented Programming and Tokens	Basics OOP Paradigm concepts with C++, Applications of C++ in variousfields.
2	**Week 3-4:** Operators, Control Statements Introduction to Classes	A simple and Structure of C++ Program. Symbols, Keywords, Identifiers, Variables and Constants which are used in C++ programs.
3	**Week 5-6:** Constructor and Destructors, Operator Overloading.	Constructors, Parameterized constructors, Multiple Constructors in a class, Constructors with default arguments, Dynamic initialization of Objects, Const object, Destructors. Introduction, Defining operator overloading,
4	**Week 7-8:** Inheritance, Polymorphism	Overloading unary and binary operators, Type Conversions.
5	**Week 9-10:** Exceptions-Handling& C++ I/O System Basics	How to use Constructors with various Parameter, Multiple Constructors in a class, Destructors. uses of constructor and application of Constructors.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies

7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50)in the semester-end examination(SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding fundamentals of C++ Programming Constructs	Students will grasp the fundamental concepts of C++ Programming, including basic constructs.
2	Executing Simple C++ Programs	Students will learn to design and execute basic and simple C++ programs.
3	Programming-Based Learning	Through program execution-based learning, students will undergo the demonstration of C++ programming constructs working principles.
4	Proficiency in C++ Specific Constructs	Students will become proficient in understanding and applying the C++ specific constructs to improve the efficiency of C++programming logics.
5	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with C++ Programming, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.
6	Understanding fundamentals of C++ Programming Constructs	Students will grasp the fundamental concepts of C++ Programming, including basic constructs.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description

M23BCS306C.1	Understand and apply the basic programming constructs.
M23BCS306C.2	Apply the structure of classes and methods in C++ programming environment.
M23BCS306C.3	Analyze the different programming constructs of C++ and its effectiveness in improving the efficiency of C++ programs.
M23BCS306C.4	Implement appropriate C++ programming constructs to solve real-world problem sample scenarios.
M23BCS306C.5	Use the concepts of input output streams for file operations

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS306C.1	**3**													
M23BCS306C.2	**3**												**3**	
M23BCS306C.3		**3**											**3**	**3**
M23BCS306C.4			**3**											
M23BCS306C.5				**3**										
M23BCS306C	**3**	**3**	**3**	**3**									**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

- Continued Evolution and Standardization: C++ continues to evolve with regular updates and new standards. The C++20 standard introduced significant new features such as modules, co routines, concepts, and improved concurrency support. Future standards, such as C++23 and beyond, are anticipated to further enhance the language, focusing on performance, simplicity, and safety. These updates ensure that C++ remains modern and relevant.

- Educational Importance: C++ continues to be a staple in computer science education. It teaches fundamental programming concepts, including memory management and system-level programming, which are essential for understanding more complex languages and systems.
- Systems and Embedded Programming: C++ is foundational in systems programming, including operating systems, drivers, and embedded systems. Its ability to interact closely with hardware while maintaining a high level of performance makes it indispensable in these areas. The Internet of Things (IoT) and smart devices will further bolster the demand for C++ in embedded systems.
- Artificial Intelligence and Machine Learning :While Python dominates the AI and machine learning space, C++ is crucial for performance-critical components of ML frameworks like TensorFlow and PyTorch. It is used to optimize algorithms and enhance the efficiency of AI applications, especially in production environments.
- Web Assembly: With the rise of Web Assembly, C++ can be used to write high-performance code that runs in the browser. This opens new avenues for C++ in web development, enabling the development of complex web applications that require near-native performance..
- Community and Ecosystem: The C++ community is vibrant and active, continually contributing to its ecosystem with libraries, tools, and frameworks. This ongoing support ensures that C++ remains relevant and accessible for developers.

3rd Semester	**Engineering Science Course (ES)** **DATA ANALYTICS with R**	**M23BCS306D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Statistics	• Understanding of fundamental statistical concepts (mean, median, mode, standard deviation).
2	Mathematics	• Proficiency in algebra and basic calculus for understanding statistical formulas and models.
3	Data Handling Concepts	• Knowledge of data types, data structures (arrays, lists, data frames), and data manipulation techniques.
4	Basic Programming Concepts	• Basic understanding of programming logic and constructs (loops, conditionals, functions).
5	R Programming Basics	• Familiarity with the R programming environment and syntax.
6	Data Visualization	• Understanding of basic data visualization principles and tools.
7	Statistical Software Usage	• Experience with any statistical software (e.g., Excel, SPSS, SAS) is beneficial but not mandatory.
8	Previous Coursework	• Completion of introductory courses in Statistics, Mathematics, or a related field.
9	Analytical Thinking	• Ability to think analytically and critically, with a focus on data interpretation and problem-solving.
10	Research Methods	• Basic knowledge of research methodologies and techniques for data collection, analysis, and interpretation.

2. Competencies

S/L	Competency	KSA Description
1	Data Manipulation in R	**Knowledge:** • Understanding of R data structures (vectors, lists, data frames). • Knowledge of data manipulation functions (e.g., dplyr, tidyr). **Skills:** • Proficiency in manipulating data using R. • Ability to clean and transform data for data preparation in analysis.
2	Statistical Analysis in R	**Knowledge:** • Understanding of basic statistical concepts (mean, median, hypothesis testing). • Knowledge of statistical functions in R. **Skills:** • Performing statistical tests and analysis using R • Interpreting statistical results. **Attitudes:** • Valuing the role of statistics in data analysis and decision-making.
3	Data Visualization with R	**Knowledge:** • Understanding of data visualization principles. • Knowledge of R visualization libraries (e.g., ggplot2). **Skills:** • Creating effective visualizations to represent data. • Customizing plots for clarity and impact. **Attitudes:** • Appreciation for the importance of clear data visualization.
4	Programming in R	**Knowledge:** • Understanding of R programming syntax and structures. • Knowledge of R programming functions and packages. **Skills:**

		• Writing efficient R scripts for data analysis. • Debugging and optimizing R code. **Attitudes:** • Openness to continuous learning and improving programming skills in R.
5	Exploratory Data Analysis (EDA)	**Knowledge:** • Understanding of EDA concepts and techniques. • Knowledge of EDA tools in R. **Skills:** • Conducting exploratory analysis to discover patterns. • Summarizing and interpreting exploratory findings. **Attitudes:** • Curiosity and initiative in exploring and understanding data.
6	Regression Analysis	**Knowledge:** • Understanding of regression concepts and models. • Knowledge of regression functions in R. **Skills:** • Performing regression analysis using R. • Interpreting regression results and diagnostics. **Attitudes:** • Appreciation for the role of regression in predictive analysis.
7	Data Cleaning and Preprocessing	**Knowledge:** • Understanding of data cleaning and preprocessing techniques. • Knowledge of tools for handling missing data and outliers in R. **Skills:** • Cleaning and preprocessing data for analysis. • Handling missing values and outliers effectively. **Attitudes:** • Valuing the importance of accurate data preparation.
8	Time Series Analysis	**Knowledge:** • Understanding of time series concepts and methods. • Knowledge of time series analysis functions in R. **Skills:** • Performing time series analysis and forecasting using R. • Interpreting time series results. **Attitudes:** • Recognition of the significance of time series analysis in various fields.
9	Machine Learning with R	**Knowledge:** • Understanding of basic machine learning concepts and algorithms. • Knowledge of machine learning libraries in R (e.g., caret). **Skills:** • Applying machine learning techniques for data analysis • Evaluating and tuning machine learning models. **Attitudes:** • Openness to applying machine learning in data projects.
10	Reporting and Communication of Results	**Knowledge:** • Understanding of effective communication principles. • Knowledge of reporting tools in R (e.g., R Markdown). **Skills:** • Creating comprehensive reports using R. • Communicating analysis results clearly and effectively. **Attitudes:** • Appreciation for clear and concise communication of data findings.

3. Syllabus

Data Analytics with R
SEMESTER – III

Course Code	**M23BCS306D**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Understand the fundamentals, standards of Functions and capabilities of R-Language.
- To know the Principles of Graphics and R-Base Graphics
- Learn how to Create/Design Interactive Graphics using R-libraries (APIs).
- To demonstrate Simulation in R-Language, Math functions and files processing.
- To Develop Applications and Evaluate Performance Using T-Tests

Module -1

INTRODUCTION OF R-LANGUAGE: Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes.

Module -2

FUNCTIONS AND STRUCTURES: R Programming Structures, Control Statements, Loops, - Looping Over Non-vector Sets, If-Else, Arithmetic and Boolean Operators, Default Values for Argument, Return Values, Functions with No Pointers in R, Recursion, Sorting and Searching.

Module -3

R-BASE GRAPHICS: Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot() Function Customizing Graphs, Saving Graphs to Files.

Module -4

T-TESTING: Probability Distributions, Normal Distribution, Binomial Distribution, Poisson Distributions, Basic Statistics, Correlation and Covariance, T-Tests, ANOVA.

Module -5

LINEAR OPTIMIZATION: Linear Models, Simple Linear Regression and Multiple Regression, Generalized Linear Models, Nonlinear Models, Splines- Decision- Random Forests.

PRACTICAL COMPONENT (Demonstration Only)

Using suitable R software /IDE, demonstrate the following programs:

1.	1a. Download and install R-Programming environment and install basic packages using install. packages () command in R. 1b. Learn all the basics of R-Programming (Datatypes, Variables Operators etc.)
2.	2a. Implement R-Loops with different examples. 2b. Learn the basic of functions in R and implement with examples.
3.	3a. Implement data frames in R. Write a program to join columns and rows in a data frame using c. bind () and **r. bind** () in R. 3b. Implement different String Manipulation functions in R.
4.	Implement different data structures in R (Vectors, Lists, Data Frames)
5.	5a. Write a program to read a .csv file and analyze the data in the file in R 5b. Create pie charts and bar charts using R
6.	Create a dataset of choice, and perform statistical analysis on the data using R
7.	7a. Write R program to find Correlation and Covariance 7b. Write R program for Regression Modeling
8.	8a. Write R program to build classification model using KNN algorithm 8b. Write R program to build clustering model using K-means algorithm

Text Books:

1. R Cookbook, Paul Teetor, Oreilly: R Cookbook [R CKBK][Paperback]Mar, 2019]
2. R in Action, Rob Kabacoff, Manning: Data Analysis and Graphics with R Mar, 2022

Reference Books:

1. “R for Everyone”, Jared P Lander, Pearson Education 2017, Latest Edition.
2. “Beginning R: An Introduction to Statistical Programming”-Larry Pace, Latest Edition.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction to R-Language	**Competency:** Understanding R-Language Fundamentals **Knowledge:** Basic math, variables, data types, advanced data structures (vectors, data frames, lists, matrices, arrays, classes). **Skills:** Running R sessions, utilizing R functions effectively.
2	Week 3-4: Functions and Structures	**Competency:** Mastering Functions and Structures in R **Knowledge:** R programming structures, control statements, loops, arithmetic and Boolean operators, recursion. Implementing loops and control structures, writing and using functions in R.
3	Week 5-6: R-Base Graphics	**Competency:** Data Visualization with R **Knowledge:** Principles of R base graphics, graph customization techniques. Creating and customizing graphs using the plot () function, saving graphs to files.
4	Week 7-8: T-Testing and Statistical Analysis	**Competency:** Conducting Statistical Tests **Knowledge:** Probability distributions, basic statistics, correlation, covariance, T-tests, ANOVA. Performing and interpreting statistical tests, conducting T-tests and ANOVA in R.
5	Week 9-10: Linear Optimization and Modeling	**Competency:** Implementing Linear and Nonlinear Models **Knowledge:** Linear and nonlinear models, simple and multiple linear regression, generalized linear models, decision trees, random forests. Building and interpreting linear and nonlinear models, implementing regression analysis.
6	Week 11-12: Integration and Practical Applications	**Competency:** Applying Learned Concepts and Competencies **Knowledge:** Apply learned concepts and competencies to real-world scenarios, hands-on practice with programming assignments, creating and interpreting visualizations, performing comprehensive data analysis.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies

7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks = (A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject **(duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding Fundamentals of R-Language	Students will grasp the fundamental concepts of R, including basic math, variables, data types, and advanced data structures such as vectors, data frames, lists, matrices, arrays, and classes.
2	Mastering Functions and Structures	Students will learn to utilize R programming structures, control statements, loops, arithmetic and Boolean operators, recursion, sorting, and searching. They will also understand how to create and use functions effectively in R.
3	Proficiency in R-Base Graphics	Students will become proficient in creating and customizing graphs using R-Base Graphics, including the use of the plot() function and saving graphs to files.
4	Conducting Statistical Tests	Students will learn to perform probability distributions, basic statistics, correlation, covariance, T-Tests, and ANOVA using R, and will develop the skills to interpret these statistical results.
5	Implementing Linear and Nonlinear Models	Students will understand and apply linear models, simple and multiple linear regression, generalized linear models, nonlinear models, splines, decision trees, and random forests for data analysis and predictive modelling in R.
6	Developing Applications and Simulations	Through practical exercises, students will simulate data, apply math functions, and process files in R. They will develop applications and evaluate their performance using T-Tests, reinforcing their understanding of theoretical concepts.
7	Creating Interactive Graphics	Students will learn to design and create interactive graphics using R libraries (APIs), enhancing their ability to visually represent data dynamically and interactively.
8	Enhancing Data Analysis Skills	Through lab sessions, students will apply their knowledge to real-world data analysis problems, including data manipulation, data structures, string manipulation, reading and analyzing CSV files, and creating visualizations like

		pie and bar charts.
9	Building and Evaluating Models	Students will write R programs for correlation, covariance, regression modelling, and build classification and clustering models using KNN and K-means algorithms, gaining hands-on experience in model development and evaluation.
10	Collaboration and Communication Skills	Students will work collaboratively on projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively, preparing them for professional teamwork environments.
11	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with data analysis, including data privacy, intellectual property rights, and adherence to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS306D.1	Understand the data in different forms
M23BCS306D.2	Apply different techniques to Explore Data Analysis and the Data Science Process
M23BCS306D.3	Analyze feature selection algorithms & design a recommender system.
M23BCS306D.4	Evaluate data visualization tools and libraries and plot graphs.
M23BCS306D.5	Develop different charts and include mathematical expressions.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS306D.1	**3**												**-**	**-**
M23BCS306D.2		**3**											**3**	**2**
M23BCS306D.3			**3**										**3**	**3**
M23BCS306D.4				**3**									**2**	**-**
M23BCS306D.5													**2**	**-**
M23BCS306D	**3**	**3**	**3**	**3**									**2.5**	**2.5**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

10.

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**

Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

Data Governance and Ethics With great power comes great responsibility. Data analysts access more sensitive and comprehensive datasets, so their role in data governance and ethics becomes more critical. Future data analysts must navigate the complexities of privacy laws, ethical data usage, and security. They will ensure that data is used responsibly, aligning with both legal requirements and moral standards.

3rd Semester	**Professional Course Lab (PCL)** **DATA STRUCTURES LABORATORY**	**M23BCSL307**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	**Fundamental Programming Skills:**	• Proficiency in at least one programming language: Often, languages like C, C++, Java, or Python are preferred. • Understanding of basic programming constructs: Loops, conditionals, functions, and recursion.
2.	**Introduction to Computer Science:**	• Basic knowledge of computer science principles: This includes understanding of algorithms, problem-solving techniques, and the basics of computational theory.
3.	**Mathematics:**	• Discrete Mathematics: Concepts like sets, graphs, combinatory, logic, and proof techniques. • Basic Algebra: Understanding of variables, equations, and functions.
4.	**Basic Algorithm Understanding:**	• Sorting and searching algorithms. • Basic algorithm design and analysis.
5.	**Specific Topics**	• Understanding of primitive data types and structures like arrays, linked lists, stacks, queues, and hash tables. • Basic understanding of memory allocation and pointers (especially for languages like C and C++). • Ability to implement recursive and iterative solutions.
6.	**Analytical Thinking:**	• Problem-solving skills: Ability to analyze problems, breaks them down into smaller parts, and devises algorithmic solutions. • Logical reasoning: Ability to understand and apply logical concepts in problem-solving.
7.	**Debugging techniques:**	• Understanding how to find and fix errors in code.

2. Competencies

S/L	Competency	KSA Description
1.	**Technical Competencies:**	**Knowledge: Programming Proficiency:** Expertise in a programming language such as C, C++, Java, or Python. **Skill:** Ability to write clean, efficient, and error-free code. Skills in organizing and prioritizing tasks effectively. **Attitude: Software Tools and IDEs:** Familiarity with development environments, debugging tools, and other software tools that aid in coding and testing.
2.	**Analytical Competencies:**	**Knowledge:** Data Structures Knowledge: Understanding of fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables, and heaps. **Skill:** Problem-Solving Skills: Ability to break down complex problems into smaller, manageable tasks. Analytical thinking to develop logical solutions. **Attitude:** Attention to Detail: Ensuring correctness in handling array indices and avoiding out-of-bound errors. Carefully displaying information to enhance user understanding.
3.	**Practical Competencies:**	**Knowledge: Memory Management:** Knowledge of dynamic memory allocation and management, especially in languages like C and C++. Understanding of concepts like pointers and references **Skill: Critical Thinking:** Capability to evaluate different approaches and choose the most effective one. Ability to debug and troubleshoot issues effectively. **Attitude: Project Management:** Ability to plan, execute, and manage projects within deadlines.

3. Syllabus

Data Structures Laboratory

SEMESTER – III			
Course Code	**M23BCSL307**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Designing, developing, implementing, and analyzing, dynamic memory management systems.
- Working with linear data structures, including stacks, queues, and linked lists, and understanding their applications.
- Understanding and applying non-linear data structures, such as trees and graphs, and exploring their various applications.

PRACTICAL COMPONENT

Implement all the programs in C Programming Language

1. Implement a menu driven Program in C for the following array operations.

- Creating an array of N Integer Elements.
- Display of array Elements with Suitable Headings.
- Inserting an Element at a given valid Position.
- Deleting an Element at a given valid Position.
- Exit.

Support the program with functions for each of the above operations.

2. Write a C program to implement the Knuth-Morris-Pratt (KMP) string matching algorithm. The program should include:

- A function to preprocess the pattern and create the match table.
- A function to search for all occurrences of the pattern in a given text using the KMP algorithm.

3. Create a menu-driven program in C to perform various operations on a stack of integers using an array-based implementation. The stack should have a maximum size defined as MAX. Perform the following operations.

- Push an Element on to Stack.
- Pop an Element from Stack.
- Demonstrate Overflow and Underflow situations on Stack.
- Display the status of Stack.
- Exit

4. Implement a Program in C for converting an Infix Expression to Postfix Expression. Program shoul support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Modulus), ^ (Power) and alphanumeric operands. Convertthis Infix Expression (a+ (b-c) *d) to Postfix Expression.

5. Create and implement a C program that provides a menu-driven interface for managing a circular queue of characters using an array-based approach. The circular queue should have a predefined maximum size, defined as MAX. Perform the following operations.

- Enqueue: Add an element to the queue.
- Dequeue: Remove an element from the queue.
- Display: Show all elements currently in the queue.

6. Create a menu-driven program in C that facilitates various operations on a singly linked list (SLL). The operations should include the following

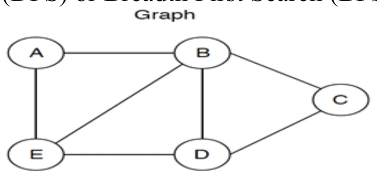
- Insertion of elements in front of the list.
- Deletion of elements in the list if it is found, otherwise display appropriate message.
- Searching an element in the list
- Display the elements of the list.

7. Create a menu-driven program in C that facilitates various operations on doubly linked list.

- Create doubly linked list of N nodes with integer data by adding each node at the front.
- Delete the node of a given data if it is found, otherwise display appropriate message.
- Insert a node to the left of the node whose key value is read as input.
- Display the contents of the list.

8. Develop a menu-driven program in C for performing various operations on a Binary Search Tree (BST) of integers. The operations include:

- Creating a BST with the following integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2

	• Traversing the BST in In-order, Pre-order, and Post-order • Searching the BST for a given element (KEY) and displaying an appropriate message • Displaying the height of the binary tree • Exiting the program
9.	Write a program in C to perform the following tasks on a graph representing N cities: • Create the graph using an adjacency matrix. • Print all the nodes reachable from a given starting node in a directed graph using either Depth First Search (DFS) or Breadth First Search (BFS) methods. 
10.	Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. • Assume that file F is maintained in memory by a Hash Table (HT) of M memory locations with L as the set of memory addresses (2-digit) of locations in HT. • Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash Function H: K%L as I (remainder method), and implement hashing techniques to map a given key K to the address space L. • Resolve the collision (if any) using linear probing.

Practice Questions to be implemented using C programming (Not be considered for exam**)**

1. Implement a dynamic array class that supports operations like adding an element, removing an element, and resizing the array when it's full.
2. Implement the binary search algorithm on a sorted array of integers.
3. Write a program to multiply two matrices and display the result.
4. Implement a stack using a linked list and perform push, pop, and display operations.
5. Write a program to implement a queue using a linked list with enqueue, dequeue, and display operations.
6. Implement a function to add two polynomials and return the result.
7. Implement a graph using an adjacency list and an adjacency matrix.
8. Write a program to perform BFS on a graph and display the traversal.
9. Implement DFS on a graph and display the traversal.
10. Implement the algorithm for constructing an optimal binary search tree and calculate the cost.

4. Syllabus Timeline

S/L	**Syllabus Timeline**	**Description**
1	Week 1-2:	Program1, Program2.
2	Week 3-4:	Program3, Program4.
3	Week 5-6:	Program5, Program6.
4	Week 7-8:	Program7, Program8.
5	Week 9-10:	Program9, Program10.
6	Week 11-12:	Practice

5. Teaching-Learning Process Strategies

S/L	**TLP Strategies:**	**Description**
1	Interactive Lectures	• Use interactive lectures to introduce new concepts. Incorporate questions and discussions to engage students.
2	Coding Sessions:	Demonstrate the implementation of data structures and algorithms live, showing step-by-step coding and debugging.
3	Lab Exercises	Design lab exercises that require students to implement and manipulate data structures.
4	Coding Assignments	Assign regular coding tasks that reinforce lecture material and provide practical experience.

5	Group Projects	Encourage students to work in groups for larger projects, fostering teamwork and collaborative problem-solving.
6	Code Documentation:	Practice writing clear and comprehensive documentation for all coding assignments and projects.

6. Assessment Details (both CIE and SEE)

Marks distribution for Program based Practical Course for CIE

Sl. No.	Description	% of Marks	In Marks
1	Observation, write-up, algorithm/program/execution	80% of the maximum	80
2	Viva-Voce	20% of the maximum	20
Total		100%	100

Marks scored by the student for 100 are scaled down to 50 marks.

SEE for practical Course (Irrespective of Experiment or program based):

1. SEE marks for practical course shall be 50 marks

Marks distribution for Experiment based Practical Course for Final CIE

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. See for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Array Operations	Understand and implement basic array operations: • Learn to create and initialize an array of integers. • Display array elements with appropriate headings. • Insert an element at a specified position in the array. • Delete an element from a specified position in the array.
2	String Matching with Knuth-Morris-Pratt (KMP) Algorithm	Develop and implement string matching algorithms: • Understand and implement the KMP algorithm for efficient string matching. • Learn to preprocess the pattern to create the partial match table. • Implement the search function to find all occurrences of the pattern in a given text.
3	Stack Operations using Array-Based Implementation	Perform various operations on a stack using arrays: • Implement push and pop operations. • Check if the stack is empty or full. • Display the elements of the stack.
4	Infix to Postfix Expression Conversion	Convert infix expressions to postfix notation: • Handle both parenthesized and non-parenthesized expressions. • Support a variety of operators including +, -, *, /, %, and ^. • Manage alphanumeric operands in expressions.
5	Circular Queue Operations using Array-Based Implementation	Manage a circular queue with array-based implementation: • Implement enqueue and dequeue operations. • Check if the queue is empty or full. • Display the elements in the circular queue.

6	Linked List Operations	Perform various operations on a singly linked list: • Insert elements at different positions. • Delete elements from the list. • Search for elements in the list. • Display the elements of the singly linked list.
7	Binary Search Tree (BST) Operations	Manage and traverse binary search trees: • Create a BST and insert elements. • Traverse the BST using in-order, pre-order, and post-order methods. • Search for a given element in the BST and display an appropriate message. • Determine and display the height of the BST.
	Dynamic Hashing	Implement dynamic hashing for efficient key-value management: • Handle insertion, deletion, and searching operations in a dynamic hash table. • Learn to dynamically resize hash tables to manage growing collections and handle collisions efficiently.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCSL307.1	Describe the fundamental concepts of basic data structures.
M23BCSL307.2	Illustrate how various data structures can be applied to solve a given problem.
M23BCSL307.3	Demonstrate the use of different data structures in practical applications.
M23BCSL307.4	Assess advanced data structures concepts, such as Hashing Techniques and Optimal Binary Search Trees, in terms of their efficiency and application.
M23BCSL307.5	Develop a solution by designing and implementing the most suitable data structure for a given problem, either individually or as part of a team.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCSL307.1	**3**												**3**	**3**
M23BCSL307.2	**3**												**3**	**3**
M23BCSL307.3		**3**											**3**	**3**
M23BCSL307.4			**3**										**3**	**3**
M23BCSL307.5			**3**	**3**									**3**	**3**
M23BCSL307	**3**	**3**	**3**	**3**									**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**5**	**10**	**10**	**5**	**20**	50
Total	**5**	**10**	**10**	**5**	**20**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**10**	**20**	**20**	**10**	**40**	100
Total	**10**	**20**	**20**	**10**	**40**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The future of data structures lab focuses on equipping students with:

Comprehensive Analysis: Develop a thorough understanding of the characteristics, advantages, and disadvantages of various linear (arrays, linked lists, stacks, queues) and non-linear (trees, graphs) data structures.

Practical Demonstration: Illustrate and implement the functioning of different data structures, showcasing their operations and practical applications in real-world scenarios.

Algorithm Application: Select and employ the most suitable searching (linear search, binary search) and sorting algorithms (bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort) based on specific problem requirements.

Real-World Problem Solving: Apply the appropriate data structure to design and implement solutions for real-world problems, ensuring efficiency, scalability, and maintainability.

By mastering these areas, students will be well-prepared to tackle advanced computer science topics and address complex challenges in software development and engineering.

3rd Semester	**Ability Enhancement Course (AE)** **SOCIAL CONNECT AND RESPONSIBILITY**	**M23BSCK308**

Social Connect and Responsibility			
Course Code	**M23BSCK308**		
Number of Lecture Hours/Week (L: T:P:S)	**(0:0:2:0)**	CIEMarks	**100**
Total Number of Lecture Hours	**24 Hours**	SEEMarks	**-**
Credits	**1**	TotalMarks	**100**

For CIE Assessment-Activities Report Evaluation by CollegeNSS Officer /HOD/SportsDept/AnyDept.

Courseobjectives:

Thiscoursewillenablestudentsto:

- Provide a formalplatform forstudentsto communicateandconnect totheirsurroundings.
- Createa responsible connection with the society.
- Understandthe community ingeneral in which theywork.
- Identify the needs and problems of the community and involvethem in problem–solving.
- Develop among themselves a sense of social & civic responsibility& utilize their knowledge in finding practical solutions to individual and communityproblems.
- Develop competence required for group living and sharing of responsibilities&gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

Contents:

- The course is mainly activity-based and will offer activities for students to connect with fellow human beings, nature, society, and the world.
- The course will engage students in interactive sessions, openmic, reading groups, story telling sessions, and semester-longactivitiesconductedbyfacultymentors.

In the following, a set of activities planned for the course have been listed:

Part I:

Plantation and adoption of a tree:

- Plantation of a tree that will be adopted for four years by a group of BE/B.Tech students. (ONE STUDENT ONE TREE).
- They will also make a nexcerpt, either as a documentary or aphotoblog, describing the plant' s origin, its usage in daily life, its appearance in folklore and literature-–Objectives, Visit, casestudy, Report, outcomes.

PartII:

Heritage walks and crafts corner:

- Heritage tour, knowing the history and culture of the city, connecting to people around through their history, knowingthe city and its craftsman, photo blog and documentary on evolution and practice of various craft forms Objectives,Visit,casestudy,Report,outcomes.

PartIII:

Organic farming and waste management:

- The usefulness of organic farming, wet waste management in neighboring villages, and implementation in the campus Objectives, Visit, casestudy, Report, outcomes.

Part IV:

Waterconservation:

- Knowing the practices in the surrounding villages and implementation in the campus, documentary orphoto blog presenting the current practices–Objectives Visit, casestudy, Report, outcomes.

PartV:

Foodwalk:

- City's culinary practices, foodlore, an dindigenous material soft heregionused in cooking–Objectives,Visit,casestudy,Report, outcomes.

Dean Academics
MIT Mysore

APPROVED
MAHARAJA INSTITUTE OF TECHNOLOGY
MIT
MYSORE
MYSORE
Dean Academics

Principal
MIT Mysore

Course outcomes (Course Skill Set)

At the end of the course, the student will be able to:

COs	Description
M23BSCK308.1	Communicate and connect to the surroundings.
M23BSCK308.2	Create a responsible connection with the society.
M23BSCK308.3	Involve in the community in general in which they work.
M23BSCK308.4	Notice the needs and problems of the community and involve them in problem–solving.
M23BSCK308.5	Develop among them selves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.
M23BSCK308.6	Develop competence required for group living and sharing responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

Activities:

The jamming session, open mic, and poetry: Platform to connect to others. Share the stories with others. Share the experience of Social Connect. Exhibit talents like playing instruments, singing, one-art play, art painting, and fine art.

Pedagogy:

The pedagogy will include interactive lectures, inspiring guest talks, field visits, social immersion, and a course project. Applying and synthesizing information from these sources to define the social problem to address and take up the solution as the course project with your group. Social immersion with NGOs/social sections will be a crucial course part.

Course topics:

The course will introduce social context and various players in the social space and present approaches to discovering and understanding social needs. Social immersion and inspiring conversations will culminate in developing an idea for problem-based intervention based on an in-depth understanding of a critical social problem.

Duration:

40 - 50 hrs engagement per semester is required for the 3rd semester of the B.E./B.Tech. Program. The students will be divided into groups. A faculty mentor will handle each group. Faculty mentors will design the activities (particularly Jamming sessions, open mic, and poetry). Faculty mentors have to design the evaluation system as per the guidelines of the scheme & syllabus.

Guideline for Assessment Process:

Continuous Internal Evaluation (CIE):

After completion of the course, the student shall prepare, with a daily diary as a reference, a comprehensive report in consultation with the mentor/s to indicate what he has observed and learned in the social connect period. The mentor should sign the Report. The Report shall be evaluated based on the following criteria and/or other relevant criteria for the completed activity. Marks allotted for the diary are out of 50. Planning and scheduling the social connect

Information/Data collected during the social connect Analysis of the information/data and report writing

Considering all the above points, allotting the marks as mentioned below

Excellent	80 to 100
Good	**60 to 79**
Satisfactory	40 to 59
Unsatisfactoryandfail	< 39

Special Note:

NO SEE–Semester End Exam–Completely Practical and activities based evaluation

Pedagogy–Guidelines:
It may differ depending onS local resources available for the study as wellas environment and climatic differences, location, andtime of execution.

Sl No	Topic	Group size	Location	Activity execution	Reporting	Evaluation of theTopic
1.	**Planation and Adoption of a tree:**	May be Individual or team	Farmers land/parks/Villages/ roadside/community area /Collegecampus etc	Site selection / Proper consultation/Continuous monitoring/Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by Faculty
2.	**Heritage walks and crafts corner:**	May be Individual or team	Temples / monumental places / Villages/ City Areas / Gramapanchayat/ public associations/ Government Schemes officers/ campus etc…	Site selection / Proper consultation/Continuous monitoring/Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluationas per the rubrics of the scheme and syllabus by Faculty
3.	**Organic farming and waste management:**	May be individual or team	Farmers land/parks/Villages visits/roadside/ community area/ College campusetc	Group selection /proper consultation / Continuous monitoring /Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by Faculty
4.	**Water conservation& Conseration Techniques:**	May beindividual or team	Villages/City Areas/Grama panchayat/public associations/ Government Schemes officers/ Campu setc…	Site selection/ Proper consultation/ Continuous monitoring/Information board	Report should be submitted by an Individual to the concerned evaluation authority	Evaluation as Per the Rubrics of the scheme and syllabus by Faculty
5.	**Food walk: Practices insociety**	May be individual or team	Villages/City Areas/Grama panchayat/ public associations/Government Schemes officers/ campusetc…	Group selection /proper consultation / Continuous monitoring /Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by Faculty

Plan of Action (Execution of Activities)

Sl. NO	**Practice Session Description**
1.	Lecture session in the field to start activities
2.	Students' Presentation on Ideas
3.	Commencement of activity and its progress
4.	Execution of Activity
5.	Execution of Activity
6.	Execution of Activity
7.	Execution of Activity
8.	Case study-based Assessment, Individual performance
9.	Sector/ Teamwise study and its consolidation
10.	Video-based seminar for10 minutes by each student at the endof these mester with a Report.

- Each student should do activities according to the scheme and syllabus.
- At the end of the semester student performance must be evaluated by the faculty for the assigned activity progress and completion.
- Atlast consolidated Repor to fall activities from 1stto5th, compiled Report should be submitted per the instructions and scheme.

AssessmentDetails:

Weightage	**CIE–100%**	
Field Visit, Plan, Discussion	**10 Marks**	• Implementation strategies of the project (NSSwork). • The last Report should be signed by the NSS Officer, the HOD, and theprincipal. • At last Report should be evaluated by the NSS officer of the institute. • Finally, the consolidated marks sheet should besent to the university and made available at the LIC visit.
Commencement of activities and its Progress	**20 Marks**	
CaseStudy-based Assessment Individual Performance with Report	**20 Marks**	
Sector-wisestudy &its consolidation 5*5 =25	**25 Marks**	
Video based seminar for10 minutes by each student At the end of semester with Report. Activities1 to 5,5*5= 25	**25 Marks**	
Tota lmarks for the course in each semester	**100 Marks**	

For each activity, 20 marks CIE will be evaluated for IA marks at the end of the semester. Report andassessmentcopy should be made availablein the department.

Students should present the progress of the activities as per the schedule in the prescribed practical session in the field.

There should be positive progress in the vertical order for society's general benefit through activities.

3rd Semester	Ability Enhancement Course (AE) DATA VISUALIZATION WITH PYTHON	M23BCS309A

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Mathematics:	Basic Mathematics is necessary
2.	Programming	Libraries: Familiarize yourself with libraries like Pandas (for data manipulation), NumPy (for numerical computations), C Programming
3.	Data Manipulation	Data Visualization: Presenting data effectively using plots, charts, and graphs.
4.	Domain Knowledge	Depending on your interests, having some knowledge in the field you want to apply Data Visualization to (e.g., finance, healthcare, marketing) can be beneficial.
5.	Critical Thinking and Problem-Solving:	Data Visualization often involves complex problems that require critical thinking and creative solutions. Practice problem-solving skills and logical reasoning.

2. Competencies

S/L	Competency	KSA Description
1.	**Understand the data**	**Knowledge:** Understanding of data, Types of data. **Skills:** Ability to Analysis the data in Real Time. Proficiency in utilizing data for Real time Application. **Attitudes:** Appreciation for the importance of data in digital system.
2.	**Visualize the data**	**Knowledge:** Understanding of Data Analytics and Visualization. **Skills:** Designing a method for Visualization using Matplotlib, Plotly and Bokeh **Attitudes:** Appreciation for the role of Data visualizationin digital systems.
3.	**Analyze the data**	**Knowledge:** Knowledge of 2D and 3D. **Skills:** Analyzing the data using Plotly Tool **Attitudes:** Valuing the importance of Real Time Data in digital system.
4.	**Refine and improve**	**Knowledge:** Knowledge of Data Analytics using different Datasets. **Skills:** Applying visualization tools for real-time Datasets **Attitudes:** Openness to learning the tools of visualization

3. Syllabus

Data Visualization with Python
SEMESTER – III

Course Code	**M23BCS309A**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Demonstrate the use of IDLE or PyCharm IDE to create Python Applications
- Using Python programming language to develop programs for solving real-world problems
- Implementation of Mat plot lib for drawing different Plots
- Demonstrate working with Seaborn, Bokeh.
- Working with Plotly for 3D, Time Series and Maps.

Sl. No.	Experiments List of problems for which student should develop program and execute in the Laboratory
1	a. Write a Python program to demonstrate how to Draw a Bar Plot using Mat plot lib. b. Write aPython program to demonstrate how to Draw a Scatter Plot using Mat plot lib.
2	a. Write a Python program to demonstrate how to Draw a Histogram Plot using Mat plot lib. b. Write a Python program to demonstrate how to Draw a Pie Chart using Mat plot lib.
3	a. WriteaPythonprogramtoillustrateLinearPlottingusingMatplotlib. b. WriteaPythonprogramtoillustrateStack and subplot usingMatplotlib.

4	WriteaPythonprogramwhichexplainsusesofcustomizingseabornplotswith Aestheticfunctions.
5	a. WriteaPythonprogramtoexplainworkingwithbokehlinegraphusingAnnotationsandLegends. b. Write a Python program for plotting different types of plots using Bokeh.
6	Write a Python program to draw 3D Plots using PlotlyLibraries.
7	a. Write a Python program to draw Time Series using Plotly Libraries. b. Write a Python program for creating Maps using Plotly Libraries.
8	For the given dataset that contains immigration details to Canada from 1980 to 2013, • Create an area plot for top 6 immigrant countries from 1990 to 2013 • Create and year-wise immigrant bar chart from India to Canada during the period of 1980 to 2013. • Create a boxplot for Indian, Phillipin and China immigrants. • Show the total no. of immigrants from India and France countries using Area Chart and Pie chart. • Create a scatter Histogram for the immigrants from Fiji and Singapore in the year 2013 https://www.kaggle.com/datasets/ammaraahmad/immigration-to-canada
9	For the given data set that contains the data of flights that were on time in January for the years 2019 and 2020. Using the two data sets visualize the data using matplot and plotly libraries to depict the following: • Show the difference in statistics for distance for both the years using the appropriate plotting technique. • Visualize the no. of flights whose destination airport id is 11778 and 11267 using a bar plot or bar chart. • Create a Sunburst Plot for both the years depicting the difference among them. • https://www.kaggle.com/datasets/divyansh22/flight-delay-prediction
10	Visualize the given Placement Data Full Class dataset that contains details about Campus Recruitment using the below techniques for appropriate dimensions and differentiate between the two techniques: • Histogram and Bar Chart [For histogram let no. of bins = 10] • Facet Plot and Pair Plot • Area Chart and Pie Chart [For yes or no data] https://www.kaggle.com/datasets/barkhaverma/placement-data-full-class
TextBooks:	
1. "Python Data Visualization Cookbook" by Igor Milovanovic, DimitryFoures, and Giuseppe Vettigli 2. "Interactive Data Visualization for the Web" by Scott Murray 3. "Data Visualization with Python and JavaScript" by Kyran Dale	
Reference Books: 1. Al Sweigart, "Automate the Boring Stuff with Python",1stEdition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) 2.Reema Thareja "Python Programming Using Problem Solving Approach" Oxford University Press. 3.Allen B. Downey, "Think Python: How to Think Like a Computer Scientist" , 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)	

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2-3-4-5-6-7: Bar, Line Scatter, Stack, Pie, Seaborn plot	**Competency**: Understand, Visualization and Analysis of Data **Knowledge:** Understanding of different Plots **Skills:** Ability to Analysis the data using different plots usingMatplotlib, Seaborn and plotly
2	Week 8-9-10: Histogram and Pie Chart	**Competency**: Visualization, Analysis and Refine of Data **Knowledge:** Understanding of Real-Time Datasets using different plots. **Skills:**

		Applying visualization tools for real-time Datasets.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of Verilog concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. **Assessment Details (both CIE and SEE)**

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20

Total	**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the importance of Data Visualize	To enable the students to become aware of different data Visualize tools and the need of Visualize.
2	Apply design Thinking	Students are able to utilize the tools for Visualize the Real time data.
3	Develop an effective interface	Student will be able to develop effective Visualization by using effective tools.
4	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS309A.1	Demonstrate the use of IDLE or PyCharm IDE to create Python Applications.
M23BCS309A.2	Use Python programming constructs to develop programs for solving real-world problems
M23BCS309A.3	Use Matplotlib for drawing different Plots
M23BCS309A.4	Demonstrate working with Seaborn, Bokeh for visualization
M23BCS309A.5	Use Plotly for drawing Time Series and Maps

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS309A.1	**3**												**3**	**3**
M23BCS309A.2		**3**											**3**	**3**
M23BCS309A.3			**3**										**3**	**3**
M23BCS309A.4					**3**								**3**	**3**
M23BCS309A.5					**3**								**3**	**3**
M23BCS309A	**3**	**3**	**3**		**3**								**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**5**	**10**	**10**	**5**	**20**	50
Total	**5**	**10**	**10**	**5**	**20**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**10**	**20**	**20**	**10**	**40**	100
Total	**10**	**20**	**20**	**10**	**40**	**100**

10. Future with this Subject

The "Data Visualization with Python" course in the Fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

- ChartGPT: ChartGPT is a revolutionary tool that uses artificial intelligence (AI) to generate data visualizations based on your text descriptions automatically. Simply tell ChartGPT what you want to see, and it will create a variety of chart options for you to choose from. It’s like having a personal data visualization assistant!
- GoodData: GoodData is a cloud-based platform that empowers businesses to build and share interactive dashboards and reports. It goes beyond basic data visualization, offering features like data modeling, security controls, and embedded analytics.
- Infogram: Infogram is a user-friendly online tool that makes creating visually stunning infographics and reports easy. It offers a vast library of dashboard templates, icons, and design elements, even for users who need design experience.
- Looker: Looker isn’t just a visualization tool; it’s a business intelligence (BI) platform that empowers data-driven decision-making. Think of it as a comprehensive data exploration hub where you can analyze, visualize, and share insights with your entire team.
- Flourish: Flourish takes a different approach, focusing on storytelling through data. It’s a tool designed to help you create compelling and interactive data stories that captivate your audience, even if they need to be data-savvy.

3rd Semester	**Ability Enhancement Course (AE)**	**M23BCS309B**

PHP PROGRAMMING

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	HTML	PHP is primarily used for web development, so having a good grasp of HTML (HyperText Markup Language) is essential. HTML is the standard markup language for creating web pages and web applications.
2.	CSS	Cascading Style Sheets (CSS) are used to style the HTML elements on a web page. Understanding CSS will help you design and layout your web pages effectively.
3.	Basic Programming Concepts:	It's beneficial to have a solid understanding of basic programming concepts such as variables, data types, loops, conditional statements, functions, and arrays. These concepts are fundamental to programming in any language, including PHP.
4.	HTTP and Web Servers:	Understanding how the web works, including concepts like HTTP (Hypertext Transfer Protocol) and how web servers operate, will give you a better understanding of how PHP scripts interact with web browsers and servers.
5.	Database Concepts: Database Concepts:	PHP is often used in conjunction with databases to create dynamic web applications. Familiarize yourself with concepts like database design, SQL (Structured Query Language), and basic database operations such as querying, inserting, updating, and deleting data.

2. Competencies

S/L	Competency	KSA Description
1.	**Web Development Basics**	**Knowledge:**HTML, CSS, and JavaScript. Understand how PHP interacts with these languages to create dynamic web pages. **Skills:** Ability to develop the web pages using HTML and CSS. Proficiency in creating webpages and interacting with PHP **Attitudes:** Appreciation for the importance of webpages in digital system.
2.	**Syntax and Language Features**	**Knowledge:** Understand PHP syntax, language constructs, data types, variables, operators, control structures (such as if statements, loops, etc.), and functions. **Skills:** Ability to write the script using PHP **Attitudes:** Confident in designing Script using PHP.
3.	**Database Integration**	**Knowledge:** Understand how to connect PHP to databases (such as MySQL, PostgreSQL, SQLite, etc.), perform database queries using SQL, and handle database transactions securely. **Skills:** Designing an user interface using PHP and DB. **Attitudes:** Designing of effective user interface.
4.	**Server-Side Scripting:**	**Knowledge:** Learn how to write PHP scripts that run on the server to generate dynamic content and interact with databases and other server-side resources. **Skills:** Writing dynamic webpage using PHP on Server side **Attitudes:** An interactive website using PHP.

3. Syllabus

PHP Programming **SEMESTER – III**			
Course Code	**M23BCS309B**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course objectives: This course will enable students to:

- To introduce the PHP syntax, elements, and control structures
- To make use of PHP Functions and File handling
- To illustrate the concept of PHP arrays and OOPs

Sl. No.	Experiments

	List of problems for which student should develop program and execute in the Laboratory
1	Create a PHP page using functions for comparing three integers and print the largest number.
2	Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3	Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
4	Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward
5	Write a PHP script that removes the whitespaces from a string. Sample string : 'The quick "" brown fox' Expected Output: Thequick""brownfox
6	Write a PHP script that finds out the sum of first n odd numbers.
7	Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
8	Write a PHP script that checks if a string contains another string.
9	Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
10	Write a simple PHP program to check that emails are valid.
11	Using switch case and dropdown list display a ―Hello message depending on the selected in drop down list.
12	Write a PHP program to print Fibonacci series using recursion.
13	Write a PHP script to replace the first 'the' of the following string with 'That'. Sample: 'the quick brown fox jumps over the lazy dog.' Expected Result: That quick brown fox jumps over the lazy dog

TextBooks:

1. Programming in HTML and PHP (Coding for Scientists and Engineers, BY DEVID R BROOKS, Springer International Publishing AG 2017
2. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.

Refernce Books:

1. David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.
2. Programming PHP, Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre,"O'Reilly Media, Inc.", 28 Apr 2006.

Tuitorials:

PHP TUTORIALS: [https://www.w3schools.com/php/}
PHP TUTORIALS: [https://www.tutorialspoint.com/php/index.htm]
HTML TUTORIALS: [https://www.w3schools.com/html/]

4.Syllabus structure

S/L	Syllabus structure	KS Description
1	Setup and Introduction	Setting up the development environment (XAMPP, WAMP, or any other preferred environment) Introduction to the lab environment and tools (IDEs, text editors) Basic PHP syntax review Simple exercises to practice writing PHP scripts
2	Working with Variables and Data Types	Hands-on exercises to reinforce understanding of variables and data types Manipulating variables and performing operationsWriting PHP scripts to demonstrate variable scope Exercises on type casting and conversion
3	Control Structures and Functions	Hands-on exercises to reinforce understanding of variables and data types Manipulating variables and performing operationsWriting PHP scripts to demonstrate variable scope Exercises on type casting and conversion
4	Database Integration	Setting up a MySQL/MariaDB database Writing PHP scripts to connect to the database Hands-on exercises on executing CRUD operations (Create, Read, Update, Delete) using PHP and MySQL/MariaDB Exercises on error handling and SQL injection prevention
5	Arrays and Forms Handling	Hands-on exercises to practice working with arrays (indexed, associative, multidimensional) Writing PHP scripts to process form data Exercises on form validation and sanitization using PHP Implementing simple CRUD operations

		with form handling

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of PHP concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. **Assessment Details (both CIE and SEE)**

Class Work: -A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

Final CIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20

Total	**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the importance of PHP	To enable the students to become aware of different Scripting Languages tools and developing website.
2	Apply design Thinking	Students are able to utilize the tools for developing the webpage.
3	Develop an effective interface	Student will be able to develop effective webpages by using PHP.
4	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS309B.1	Apply basic concepts of PHP to develop web program
M23BCS309B.2	Develop programs in PHP involving control structures
M23BCS309B.3	Develop programs to handle structured data (object) and data items (array)
M23BCS309B.4	Develop programs to access and manipulate contents of files
M23BCS309B.5	Use super-global arrays and regular expressions to solve real world problems.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS309B.1	3												3	3
M23BCS309B.2		3											3	3
M23BCS309B.3			3										3	3
M23BCS309B.4					3								3	3
M23BCS309B.5					3								3	3
M23BCS309B	3	3	3		3								3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**5**	**10**	**10**	**5**	**20**	50
Total	**5**	**10**	**10**	**5**	**20**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**10**	**20**	**20**	**10**	**40**	100
Total	**10**	**20**	**20**	**10**	**40**	**100**

10. Future with this Subject

The "PHP Programming" course in the Third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

PHP 8.x Features and Beyond

- JIT (Just-In-Time) Compilation: PHP 8 introduced JIT, which can significantly improve performance for certain types of applications. Understanding JIT and how to optimize code for it is essential.
- New Syntax and Features: PHP 8.x brought new syntax, such as union types, attributes (annotations), named arguments, and match expressions. Future versions will continue to add features that can streamline development.

Frameworks and Tools

- Laravel: Laravel continues to be one of the most popular PHP frameworks. Its ecosystem and tools likeLaravel Vapor (for serverless applications) are important subjects.
- Symfony: Another major framework that is continually evolving. Knowing the latest updates and best practices for Symfony is crucial.
- Other Frameworks: Understanding other frameworks like CodeIgniter, Yii, and Zend Framework (Laminas) can also be beneficial.

API Development

- RESTful APIs: Building and consuming RESTful APIs is a key skill, with tools like Lumen (a micro-framework by Laravel) being useful.
- GraphQL: An alternative to REST, GraphQL is gaining traction. Learning how to implement GraphQL in PHP applications can be valuable.

Security

- Modern Security Practices: Emphasis on securing PHP applications, including understanding common vulnerabilities (like SQL injection, XSS, CSRF) and how to mitigate them.
- Security Tools: Familiarity with tools and libraries that aid in securing PHP applications.

Testing and Quality Assurance

- Unit Testing: Using PHPUnit to write and run unit tests to ensure code quality.
- Integration and Functional Testing: Tools like Behat for behavior-driven development (BDD).

DevOps and Continuous Integration/Continuous Deployment (CI/CD)

- CI/CD Pipelines: Setting up automated pipelines using tools like GitHub Actions, GitLab CI, or Jenkins.
- Containerization: Using Docker to containerize PHP applications for consistent development and production environments.

Performance Optimization

- Caching Strategies: Using tools like Redis or Memcached to improve application performance.
- Profiling and Monitoring: Tools like Xdebug, Blackfire, or New Relic for profiling and monitoring applications.

Modern Development Practices

- Composer: Dependency management with Composer, understanding how to create and manage packages.PSR Standards: Adhering to PHP-FIG PSR standards (like PSR-12 for coding style).

3rd Semester	**Ability Enhancement Course (AE)** **PROJECT MANAGEMENT WITH REPORT WRITING**	**M23BCS309C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Computer Skills	**Familiarity with File Management**: Ability to manage files and folders on a computer. **Basic Software Skills**: Comfortable using web browsers and common software applications. **Typing Proficiency**: Ability to type efficiently and use keyboard shortcuts.
2	Basic Understanding of Document Preparation	**Word Processing**: Basic knowledge of using word processors like Microsoft Word or Google Docs. **Formatting Documents**: Understanding of basic document formatting, such as setting margins, headings, and paragraphs.
3	Fundamental Mathematical Knowledge	**Mathematical Notation**: Familiarity with basic mathematical symbols and notation. **Basic Algebra and Calculus**: Understanding of fundamental concepts in algebra and calculus, as these are often needed for typesetting equations.
4	Introduction to Programming or Markup Languages	**Basic Programming Concepts**: Familiarity with basic programming or scripting concepts can be beneficial but is not mandatory. **Markup Languages**: Understanding of HTML, Markdown, or other markup languages can be helpful for grasping LaTeX syntax.
5	Basic Knowledge of Bibliographies and Citations	**Understanding Citations**: Familiarity with the purpose and format of citations and references. **Bibliography Management**: Basic knowledge of bibliography management tools like EndNote, Zotero, or Mendeley
6	English Proficiency	**Reading and Writing**: Proficiency in reading and writing English, as most of the documentation and tutorials for Overleaf and LaTeX are in English.

2. Competencies

S/L	Competency	KSA Description
1	Document Creation and Management	**Knowledge:** Understand how to create, upload, copy, and manage projects in Over leaf. **Skills:** Create new documents and projects from scratch or using templates. Upload existing LaTeX projects to Overleaf and copy projects for duplication. Insert and correctly position images within documents. **Attitudes:** Adaptability to work in different environments, including offline, and to use collaborative tools effectively.
2	Proficient Mathematical Typesetting Knowledge	**Knowledge:** Understand the syntax and commands for writing mathematical expressions in LaTeX. **Skills:** Apply subscripts and superscripts correctly to denote mathematical notations. Use brackets and parentheses effectively to group expressions. **Attitudes:** Diligence in ensuring that mathematical content is presented clearly and professionally.
3	Effective Use of Mathematical Fonts and Symbols	**Knowledge:** Understand the different mathematical fonts available in LaTeX and their applications. **Skills:** Apply appropriate mathematical fonts to enhance the presentation of equations and symbols. Adjust spacing in math mode to ensure mathematical expressions are well-formatted and easy to read. **Attitudes:** Openness to experimenting with different styles and fonts to find the most effective presentation for mathematical content.

4	Effective Management of References and Citations	**Knowledge:** Understand the principles and methods of bibliography management using BibTeX, natbib, and biblatex. **Skills:** Manage bibliographic data efficiently using BibTeX, natbib, or biblatex, depending on the requirements of the project. Apply various bibliography styles to format the list of references according to specific citation guidelines. Use citation styles to insert and format in-text citations accurately within the document. **Attitudes:** Thoroughness in verifying and cross-checking references to ensure accuracy and completeness.
5	Multilingual Typesetting and Document Structure	**Knowledge:** Understand the differences between polyglossia and babel packages for multilingual typesetting in LaTeX. **Skills:** Configure Overleaf documents for multilingual typesetting using polyglossia or babel along with fontspec. Apply appropriate font settings and language-specific features to accurately represent different languages. **Attitudes:** Precision in managing cross-references and maintaining consistency across language versions of documents.
6	Advanced Document Formatting and Presentation in LaTeX	**Knowledge:** Understand the different aspects of formatting in LaTeX, including setting lengths, headers, footers, and page numbering. Understand the use of counters, footnotes, and margin notes in document formatting. **Skills:** Set and adjust lengths, headers, footers, and page numbering to meet specific formatting requirements. Select and apply appropriate font sizes, families, styles, and typefaces to enhance document readability and presentation. **Attitudes:** Creativity in selecting fonts and designing layouts to enhance the visual appeal of documents and presentations. Openness to experimenting with different formatting options and presentation tools to find the best solutions for each project.

3. Syllabus

Project Management with Report Writing
SEMESTER – III

Course Code	**M23BCS309C**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(1:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Learn to create, upload, copy, and manage documents in Overleaf.
- Typeset complex math expressions, including subscripts, superscripts, fractions, binomials, and aligned equations.
- Typeset complex math expressions, including subscripts, superscripts, fractions, binomials, and aligned equations.
- Manage bibliographies using BibTeX, natbib, and biblatex.

Module -1

Creating a document in Overleaf

Uploading a project, Copying a project, Creating a project from a template, Using the Overleaf project menu, Including images in Overleaf, Exporting your work from Overleaf, Working offline in Overleaf, Using Track Changes in Overleaf, Using bibliographies in Overleaf, Sharing your work with others, Using the History feature.

Module -2

Mathematics

Mathematical expressions, Subscripts and superscripts, Brackets and Parentheses, Matrices, Fractions and Binomials, aligning equations, Operators, spacing in math mode, Integrals, sums and limits, display style in math mode, List of Greek letters and math symbols, Mathematical fonts, Using the Symbol Palette in Overleaf.

Module -3

Figures and tables

Inserting Images, Tables, Positioning Images and Tables, Lists of Tables and Figures, Drawing Diagrams, Directly in LaTeX

References and Citations

Bibliography management with bibtex, Bibliography management with natbib, Bibliography management with biblatex, Bibtex bibliography styles, Natbib bibliography styles, Natbib citation styles, Biblatex bibliography styles, Biblatex citation styles.

Module -4

Languages: Multilingual typesetting on Overleaf using polyglossia and fontspec, Multilingual typesetting on Overleaf using babel and fontspec, international language support, Quotations and quotation marks.

Document structure

Sections and chapters, Table of contents, Cross referencing sections, equations and floats, Indices, Glossaries, Nomenclatures, Multi-file LaTeX projects, Hyperlinks.

Module -5

Formatting

Lengths in LaTeX, Headers and footers, Page numbering, Paragraph formatting, Line breaks and blank spaces, Text alignment, Page size and margins, Single sided and double sided documents, Multiple columns, Counters, Footnotes, Margin notes

Fonts

Font sizes, families, and styles, Font typefaces, Supporting modern fonts with XƎLaTeX

Presentations

Beamer, Powerdot, Posters,

Commands

CommandsEnvironments

Field specific

Theorems and proofs, Chemistry formulae, Feynman diagrams, Molecular orbital diagrams Chess notation.

Reference: https://www.overleaf.com/learn (details for all individual topics are available at this link)

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1: Introduction to Overleaf and Basic Document Creation	**Topics**: Creating a document, uploading a project, copying a project, using the Overleaf project menu **Activities**: Hands-on practice with creating and managing projects
2	Week 2: Working with Text and Formatting	**Topics**: Text alignment, paragraph formatting, headers and footers, page numbering **Activities**: Formatting assignments
3	Week 3: Including Images and Tables	**Topics**: Inserting images, positioning images and tables, lists of tables and figures **Activities**: Assignments involving adding and formatting images and tables
4	Week 4: Mathematical Expressions and Symbols	**Topics**: Mathematical expressions, subscripts and superscripts, brackets and parentheses, matrices **Activities**: Math-heavy document creation
5	Week 5: Advanced Mathematical Features	**Topics**: Fractions, binomials, aligning equations, integrals, sums, and limits **Activities**: Advanced math document creation
6	Week 6: Bibliographies and Citations	**Topics**: Bibliography management with bibtex, natbib, biblatex; citation styles **Activities**: Creating documents with bibliographies
7	Week 7: Document Structure and Cross-Referencing	**Topics**: Sections and chapters, table of contents, cross-referencing **Activities**: Creating structured documents

8	Week 8: Multilingual Typesetting and Advanced Formatting	**Topics**: Polyglossia, babel, international language support, page size and margins **Activities**: Multilingual document creation
9	Week 9: Figures, Tables, and Diagrams	**Topics**: Drawing diagrams directly in LaTeX, lists of tables and figures **Activities**: Creating complex figures and tables
10	Week 10: Presentations and Field-Specific Features	**Topics**: Beamer presentations, theorems and proofs, chemistry formulae **Activities**: Creating presentations and field-specific documents
11	Week 11: Working Offline and Exporting Documents	**Topics**: Exporting work, working offline, using the History feature **Activities**: Offline work and document export exercises
12	Week 12: Review and Final Project	**Activities**: Final project encompassing all learned features, review sessions, and peer feedback

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Sequential Learning**	Begin with basic concepts like creating a document, uploading a project, and copying a project.
2	Step-by-Step Tutorials	Gradually introduce more complex topics like mathematical expressions, figures, tables, and bibliographies.
3	**Interactive Sessions**	Conduct live coding sessions where students follow along on their own Overleaf accounts.
4	Assignments	Give practical assignments requiring the use of specific Overleaf features (e.g., creating a document with a bibliography and figures)
5	**Projects**	Have students work on larger projects that incorporate multiple Overleaf and LaTeX features.
6	Overleaf Documentation	Use Overleaf’s comprehensive documentation as a textbook.
7	Templates	Provide students with templates to help them get started and understand the structure of LaTeX documents.
8	**Online Resources**	Utilize online forums, tutorials, and videos for additional learning materials.
9	Group Work	Assign group projects to encourage collaboration and peer learning.
10	Project-Based Learning	Engage in projects that require comprehensive use of LaTeX and Overleaf.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation (CIE):

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks (A+B)			**50**	**20**

The CIE question paper shall have MCQ set for 25 questions, each carrying one mark.
Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination: Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.

- There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.

- The students have to answer 5 full questions selecting one full question from each module.
- Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understand the Overleaf Interface**	Identify and navigate the main components of the Overleaf interface. Explain the functions of the Overleaf project menu.
2	**Create and Manage Projects**	Create a new project from scratch and from templates. Upload and copy existing projects into Overleaf. Share projects with collaborators and manage permissions.
3	**Typeset Mathematical Expressions**:	Write and format mathematical expressions and equations. Use subscripts, superscripts, brackets, and parentheses.
4	**Manage Lists and Draw Diagrams**:	Generate lists of tables and figures. Draw and integrate diagrams directly in LaTeX.
5	**Organize and Structure Documents**	Create sections, chapters, and a table of contents. Cross-reference sections, equations, and floats. Manage indices, glossaries, and nomenclatures.
6	**Create Presentations**	Design and create presentations using Beamer and Powerdot. Develop posters using LaTeX.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS309C.1	Apply knowledge to create, manage, and export documents in Overleaf, including uploading existing projects, utilizing templates, and working offline.
M23BCS309C.2	Analyze and construct complex mathematical content using LaTeX, including expressions, equations, matrices, and symbols
M23BCS309C.3	Design lists of tables and figures, and develop diagrams directly within LaTeX to enhance the visual structure of documents.
M23BCS309C.4	Evaluate and implement effective bibliography and citation management using bibtex, natbib, and biblatex. Critically apply different bibliography and citation styles to ensure accurate and consistent referencing in academic documents.
M23BCS309C.5	Synthesize multilingual typesetting techniques using polyglossia, babel, and fontspec. Organize and structure documents with sections, chapters, cross-referencing, indices, glossaries, and hyperlinks.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS309C.1	3	2	2	-	3	-	-	-	2	3	2	3	3	2
M23BCS309C.2	3	3	2	3	3	-	-	-	-	2	2	3	3	3
M23BCS309C.3	3	2	3	-	3	-	-	-	2	2	3	2	3	2
M23BCS309C.4	3	3	2	3	3	-	-	-	2	2	2	3	3	3
M23BCS309C.5	3	2	3	2	3	-	-	-	2	3	3	3	3	2
M23BCS309C	3	2.4	2.4	2.6	3				2	2.6	2.4	2.8	3	2.4

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:
Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The future of this syllabus, which focuses on Overleaf and LaTeX for creating and managing scientific documents, mathematical typesetting, and advanced document formatting, looks promising due to several key trends and developments in academia, industry, and technology. Here are some predictions and potential directions for the future:

- **Real-Time Collaboration:** The demand for real-time collaborative tools will grow, with Overleaf continuing to enhance features that support remote teamwork, making it indispensable for global research collaborations.
- **Cloud-Based Solutions:** Overleaf's cloud-based platform will align well with the trend towards cloud computing, enabling seamless access and collaboration from anywhere.
- **Artificial Intelligence:** AI and machine learning could be integrated into Overleaf to offer advanced features such as auto-completion of code, error detection, and suggestions for improvements in document formatting and content.
- **Enhanced Usability:** Continuous improvements in user interface and experience will make Overleaf more accessible to beginners, reducing the learning curve associated with LaTeX.
- **Interdisciplinary Applications:** The syllabus can be expanded to include more interdisciplinary applications, such as data science, engineering, social sciences, and humanities, where Overleaf and LaTeX can be used for complex document preparation.
- Professional and Corporate Use: Beyond academia, the use of LaTeX in professional reports, technical documentation, and corporate publications will increase due to its superior handling of complex formatting and typesetting.
- **Curriculum Integration:** Educational institutions may integrate this syllabus more deeply into their curricula, not just in STEM fields, but also in fields that require rigorous documentation and publication standards.
- **Online Learning and MOOCs:** The syllabus can be adapted for online courses and Massive Open Online Courses (MOOCs), making advanced document preparation skills widely accessible.
- **Community Development:** The open-source nature of LaTeX will continue to benefit from community contributions, leading to the development of new packages and features that expand its capabilities.
- **Collaborative Platforms:** Platforms like Overleaf will continue to evolve through user feedback and community contributions, enhancing their functionality and user experience.

- **Sustainable Practices:** Emphasis on sustainable practices in digital documentation can lead to more efficient and eco-friendly ways of managing and sharing documents.
- **Accessibility Enhancements:** Improved accessibility features will make Overleaf and LaTeX more usable for individuals with disabilities, promoting inclusivity in document preparation.

3rd Semester	**Ability Enhancement Course (AE)** **DATA ANALYTICS WITH EXCEL**	**M23BCS309D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Excel skills	You should be comfortable with navigating Excel, entering data, formatting cells, using formulas and functions, and creating basic charts and tables.
2.	Understanding of data types	Knowing the different types of data (e.g., numerical, text, date/time) and how Excel handles them is essential for data analysis
3.	Knowledge of Excel functions	* Understanding commonly used functions like SUM, AVERAGE, COUNT, IF, VLOOKUP, INDEX/MATCH, and pivot tables will be beneficial for data analysis tasks
4.	Data cleaning skills	Being able to clean and prepare data for analysis, including dealing with missing values, duplicates, and inconsistencies, is crucial for accurate analysis.
5.	Basic statistical concepts	Familiarity with basic statistical concepts like mean, median, mode, standard deviation, and correlation will help you interpret and analyze data effectively
6.	Pivot tables	Knowing how to create and manipulate pivot tables is essential for summarizing and analyzing large datasets in Excel
7.	Data visualization	Understanding how to create effective data visualizations such as charts and graphs to communicate insights from your analysis.
8.	Data analysis tools	Familiarity with Excels data analysis tools, such as regression analysis, scenario analysis, and data modeling, can enhance your analytical capabilities.

2. Competencies

S/L	Competency	KSA Description
1.	**Data Cleaning and Preparation**	**Knowledge**: Understanding Excel's tools, functions, and techniques for data cleaning Being able to clean and prepare data for analysis is crucial. This involves tasks such as removing duplicates, handling missing values, and formatting data properly **Skills**: The abiity to apply Excel effectively for data management, analysis, reporting, and communication, using both basic and advanced Excel features. **Attitudes**: Cultivating the right mindset—attention to detail, curiosity, adaptability, collaboration, and ethical practices—that enhances the effectiveness of data analysis.
2.	**Formulas and Functions**	**Knowledge**: Excel has a wide range of built-in functions that can be used for data analysis, such as SUM, AVERAGE, VLOOKUP, IF statements, and more. Understanding how to use these functions effectively is essential **Skills**: Building custom formulas, combining multiple functions (e.g., **IF** + **AND** + **VLOOKUP**) to perform complex calculations.. **Attitudes**: Careful checking of formulas to avoid small errors, such as incorrect references or missing parentheses, which can lead to incorrect results
3.	**Pivot Tables**	**Knowledge**: Understanding Excel's tools, functions, and techniques for data analysis, visualization, and statistical analysis. **Skills**: The ability to apply Excel effectively for data management, analysis, reporting, and communication, using both basic and advanced Excel features. **Attitudes**: Pivot Tables allow you to summarize and analyze large datasets easily. Knowing how to create Pivot Tables, customize them, and interpret the results is a key skill in Excel data analytics.
4.	**Data Visualization**	**Knowledge**:. Excel offers various tools for data visualization, such as charts and graphs. Being able to create clear and informative visualizations to present your findings is important **Skills**: pplying statistical techniques such as **ANOVA**, **t-tests**, and **chi-square tests** using Excel's **Analysis ToolPak**. **Attitudes**: A proactive mindset for experimenting with different Excel functions, tools, and techniques to uncover insights and solve business problems.

5.	**Statistical Analysis**	**Knowledge**: Excel also provides capabilities for basic statistical analysis, such as calculating mean, median, mode, standard deviation, and performing regression analysis. Understanding statistical concepts and how to apply them in Excel is valuable. **Skills**: Performing **descriptive statistics** using built-in Excel functions (e.g., mean, median, mode, standard deviation) to analyze data distributions. **Attitudes**: Continuously looking for ways to automate tasks (e.g., using **Macros** or **VBA**) and improving workflows to reduce manual effort.
6.	**Data Tables and What-If Analysis**	**Knowledge**: Excel allows for creating data tables and performing what-if analysis to explore different scenarios and understand the impact of changes in variables on outcomes **Skills**: Conducting **What-If Analysis** using **Goal Seek, Scenario Manager, and Data Tables to perform predictive modeling.** **Attitudes**: Ensuring data privacy, avoiding bias in analysis, and maintaining transparency when presenting findings.
7.	**Data Mining**	**Knowledge**: . Excel can be used for simple data mining tasks, such as finding patterns, trends, and relationships in data. Skills in sorting, filtering, and grouping data are essential for data mining in Excel **Skills**: Applying **Data Validation** rules to standardize data inputs (e.g., ensuring dates are in a consistent format or setting ranges for numerical inputs). **Attitudes**: Continuously looking for ways to automate tasks (e.g., using **Macros** or **VBA**) and improving workflows to reduce manual effort.
8.	**Data Visualization Techniques**	**Knowledge**: Knowledge of various data visualization techniques such as histograms, scatter plots, box plots, and heat maps can help in presenting data in a meaningful way. **Skills**: The ability to apply Excel effectively for data management, analysis, reporting, and communication, using both basic and advanced Excel features. **Attitudes**: Cultivating the right mindset—attention to detail, curiosity, adaptability, collaboration, and ethical practices—that enhances the effectiveness of data analysis.
9.	**Data Interpretation**	**Knowledge**: Being able to interpret the results of your analysis and draw meaningful insights from the data is a crucial competency in data analytics with Excel **Skills**: The ability to apply Excel effectively for data management, analysis, reporting, and communication, using both basic and advanced Excel features. **Attitudes**: Cultivating the right mindset—attention to detail, curiosity, adaptability, collaboration, and ethical practices—that enhances the effectiveness of data analysis.
10.	**Automation and Efficiency**	**Knowledge**: Excel offers ways to automate repetitive tasks and increase efficiency, such as using macros and VBA programming. Understanding how to automate processes can save time and improve productivity in data analysis. **Skills**: The ability to apply Excel effectively for data management, analysis, reporting, and communication, using both basic and advanced Excel features. **Attitudes**: Cultivating the right mindset—attention to detail, curiosity, adaptability, collaboration, and ethical practices—that enhances the effectiveness of data analysis.

3. Syllabus

<table>
<tr><th colspan="4">DATA ANALYTICS WITH EXCEL
SEMESTER – III</th></tr>
<tr><td>Course Code</td><td><b>M23BCS309D</b></td><td>CIE Marks</td><td><b>50</b></td></tr>
<tr><td>Number of Lecture Hours/Week(L: T: P: S)</td><td><b>(0:0:2:0)</b></td><td>SEE Marks</td><td><b>50</b></td></tr>
<tr><td>Total Number of Lecture Hours</td><td><b>24 Hours</b></td><td>Total Marks</td><td><b>100</b></td></tr>
<tr><td>Credits</td><td><b>01</b></td><td>Exam Hours</td><td><b>03</b></td></tr>
<tr><td colspan="4"><b>Course objectives:</b> This course will enable students to:
• To Apply analysis techniques to datasets in Excel
• Learn how to use Pivot Tables and Pivot Charts to streamline your workflow in Excel
• Understand and identify the principles of data analysis</td></tr>
</table>

- Become adept at using Excel functions and techniques for analysis
- Build presentation ready dashboards in Excel.

Sl. NO	Experiments
1	**Getting Started with Excel:** Create a student database with following fields student usn, name, department, course, semester and perform the following operations Insertion of rows and columns, Drag & Fill, use of Aggregate functions...
2	**Working with Data**: Create a student database with following fields student usn, name, department, course, semester and perform the following operations Importing data, Data Entry & Manipulation, Sorting & Filtering.
3	**Working with Data:** Create a student database with following fields student usn, name, department, course, semester and perform the following operations Data Validation, Pivot Tables & Pivot Charts
4	**Data Analysis Process:** Create a student database with following fields student usn,name,department,course,semester and perform the following operations Conditional Formatting, What-If Analysis, Data Tables, Charts & Graphs
5	**Cleaning Data with Text Functions:** Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP).use of UPPER and LOWER, TRIM function, Concatenate.
6	**Cleaning Data Containing Date and Time Values:** Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP).use of DATEVALUE function, DATEADD and DATEDIF, TIMEVALUE functions.
7	**Conditional Formatting:** Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP). formatting, parsing, and highlighting data in spreadsheets during data analysis
8	**Working with Multiple Sheets:** work with multiple sheets within a workbook is crucial for organizing and managing data, perform complex calculations and create comprehensive reports.
9	Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP). Use appropriate formulas to calculate the above scenario. Analyse the data using appropriate chart and report the data
10	Create worksheet on Inventory Management: Sheet should contain Product code, Product name, Product type, MRP, Cost after % of discount, Date of purchase. Use appropriate formulas to calculate the above scenario. Analyse the data using appropriate chart and report the data.
11	Create worksheet on Sales analysis of Merchandise Store: data consisting of Order ID, Customer ID, Gender, age, date of order, month, online platform, Category of product, size, quantity, amount, shipping city and other details. Use of formula to segregate different categories and perform a comparative study using pivot tables and different sort of charts.
12	Generation of report & presentation using Autofilter&macro.

Text Books:

1. Berk& Carey - Data Analysis with Microsoft® Excel: Updated for Office 2007®, Third Edition, © 2010 Brooks/Cole, Cengage Learning, ISBN-13: 978-0-495-39178-4

2.Wayne L. Winston - Microsoft Excel 2019: Data Analysis And Business Modeling, PHI, ISBN: 9789389347180

Reference Books:

1. Aryan Gupta - Data Analysis in Excel: The Best Guide. (https://www.simplilearn.com/tutorials/excel-tutorial/data-analysis-excel)

2.Comprehensive & Easy Guide to Learn Advanced MS Excel: By Naveen Mishra,

Course outcomes (Course Skill Set):

At the end of the course the student will be able to:

● Use advanced functions and productivity tools to assist in developing worksheets.

● Manipulate data lists using Outline and PivotTables.

● Use Consolidation to summarize and report results from multiple worksheets.

- Apply Macros and Autofilter to solve the given real world scenario.

4. Syllabus Timeline

S/L	Syllabus Timeline	Competency (KSA) Developed
1	Lab 1:Getting Started with Excel	Install & Configure: Knowledge of Excel worksheet Skills: students will know the installation process Attitudes: Problem Solving, Documentation
2	Lab 2: Working with Data	Knowledge of Data Modeling, Skills: Create, Read, Update, Delete. Attitudes: Organization, Attention to Detail
3	Lab 3: Working with Data	Knowledge of Pivot table and charts Skills: gain knowledge to slove Complex Queries. Attitudes: improves Critical Thinking, Innovation
4	Lab 4: Data Analysis Process	Knowledge of what if analysis, Skills: tables, charts, Average Data. Attitudes: Analytical Thinking
5	Lab 5: Cleaning Data with Text Functions	Index Mastery: Knowledge of trim functions, Skills: Creating & Utilizing trim, Performance Comparison. Attitudes: Efficiency, Optimization
6	Lab 6: Cleaning Data Containing Date and Time Value	Data Integrity: Knowledge of Schema Design, Validation Rules, Skills: Implementing Validation, Error Handling. Attitudes: Accuracy, Best Practices
7	Lab 7: Conditional Formatting	Knowledge of conditional formatting, Skills: formatting, parsing, Data Insertion. Attitudes: Adaptability, Exploration
8	Lab 8: Working with Multiple Sheets	Knowledge of User Roles, Permissions, Skills: User Creation, Role Assignment. Attitudes: Security-Mindedness, Responsibility
9	Lab 9: Create worksheet	Knowledge of worksheet creation, Skills: various excel Operations, Attitudes: Problem Solving, Resourcefulness
10	Lab 10: Create worksheet on Inventory Management	Knowledge of EXcel Functions, Skills: Developing Functions. Attitudes: Innovation, Adaptability
11	Lab 11: Create worksheet on sales Management	Project Presentation and Review
12	Lab 12: Evaluation	Usage of MAcros

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Demonstrations	Interactive Demonstrations The instructor demonstrates key Excel concepts and commands. Students follow along and can ask questions during the demonstration. This helps build a foundation of understanding and allows for immediate clarification.
2	Hands-on Exercises	Students complete practical exercises based on the lab objectives. These exercises can range from basic operations to more complex queries and aggregation pipelines. This approach provides active learning and reinforces the concepts learned.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Case Studies and RealWorld Examples	Case Studies and RealWorld Examples The instructor presents realworld scenarios where MongoDB is used. This helps students understand the practical applications of MongoDB and how it solves real-world problems.

5	Project-Based Learning	Students work on a larger project that requires them to apply the learned MongoDB concepts and skills. This approach encourages deeper learning, creativity, and problem-solving.

6. **Assessment Details (both CIE and SEE)**

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. **Learning Objectives**

S/L	Learning Objectives	Description
1	Understanding Fundamentals To Apply analysis techniques to datasets in Excel	This objective sets the stage for the course, introducing students to the core principles of Excel. It provides context for the upcoming lab programs.
2	Learn how to use Pivot Tables and Pivot Charts to streamline your workflow in Excel	This objective emphasizes the practical skills students will acquire in the initial lab programs. It prepares them for more complex tasks and data analysis in later labs.
3	Understand and identify the principles of data analysis	This objective highlights the importance of learning advanced data analysis techniques. These skills are essential for extracting valuable insights from data.
4	Become adept at using Excel functions and techniques for analysis	This objective underscores the practical aspects of working with large datasets in Microsoft Excel. Students will learn to optimize data access and maintain data quality

5	Build presentation ready dashboards in Excel	This objective aims to introduce students to the real-world applications of Microsoft excel. They will learn how to secure their data and manage access to cloud-based databases.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS309D.1	Employing productivity tools and advanced functions to help with worksheet creation and development.
M23BCS309D.2	Utilizing outline pivot tables, and manipulating data listings.
M23BCS309D.3	Use consolidation to compile and report findings from multiple worksheets,
M23BCS309D.4	Apply macros and auto filters to solve the real-world scenario

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS309D.1	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**2**
M23BCS309D.2	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**-**
M23BCS309D.3	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**-**
M23BCS309D.4	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**
M23BCS309D	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2.33**	**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**10**	**10**	**10**	**20**		**50**
Total	**10**	**10**	**10**	**20**		**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 12	**20**	**20**	**20**	**40**		**100**
Total	**20**	**20**	**20**	**40**		**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

There are several future topics you can consider learning based on Excel to enhance your data analytics skills. Here are a few suggestions:

- **Advanced Formulas and Functions**: Delve into more complex Excel formulas and functions like INDEX-MATCH, SUMIFS, and VLOOKUP to manipulate and analyze data more effectively.
- **Pivot Tables and Pivot Charts**: Mastering PivotTables and PivotCharts can help you summarize, analyze, and visualize large data sets in Excel efficiently.
- **Data Visualization Techniques**: Explore Excel's data visualization capabilities, such as creating interactive dashboards, custom charts, and graphs to effectively communicate insights from data.
- **Data Analysis Tools**: Learn how to use Excel's built-in data analysis tools like Goal Seek, Solver, and Data Tables for advanced analytics and problem-solving.
- **Data Cleaning and Data Preparation**: Focus on techniques for cleaning and preparing data in Excel, including removing duplicates, handling missing values, and transforming data for analysis.
- **Macro Programming with VBA**: If you're interested in automation, consider learning Visual Basic for Applications (VBA) to create macros and automate repetitive tasks in Excel.

- **Statistical Analysis in Excel**: Explore statistical functions and tools in Excel to perform common statistical analyses like regression, hypothesis testing, and ANOVA.
- **Integration with Other Tools**: Learn how to integrate Excel with other data analytics tools like Power Query, Power Pivot, and Power BI for more advanced data analysis tasks.

3rd Semester	**National Service Scheme /Physical Education/Yoga (NCMC)** **NATIONAL SERVICE SCHEME (NSS)**	**M23BNSK310**

National Service Scheme (NSS)			
Course Code	**M23BNSK310/410/509/608**		
Number of Lecture Hours/Week(L:T:P:S)	0:0:2:0	CIEMarks	100
Total Number of Lecture Hours		SEEMarks	**-**
Credits	**0**	TotalMarks	100
Activities Report Evaluation by College NSS Officer at the end of every semester (3rd to 6th semester)			

Course Objectives:

National Service Scheme (NSS) will enable students to:

1. Understand the community in general in which they work.
2. Identify the needs and problems of the community and involve them in problem–solving.
3. Develop among them a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.
4. Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.
5. Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

General Instructions-Pedagogy:

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. Inaddition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the activities will develop students' theoretical and applied social and cultural skills.
2. State the need for NSS activities and its present relevance in the society and provide real-life examples.
3. Support and guide the students for self-planned activities.
4. You will also be responsible for assigning home work, grading assignments and quizzes, and documenting students' progress in real activities in the field.
5. Encourage the students for group work to improve their creative and analytical skills.

Contents:

1. Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing.
2. Waste management–Public, Private and Govt organization, 5R's.
3. Setting of the information imparting club for women leading to contribution in social and economic issues.
4. Water conservation techniques–Role of different stake holders–Implementation.
5. Preparing an action able business proposal for enhancing the village income and approach for implementation.
6. Helping local schools to achieve good results and enhance their enrolmentin Higher/technical/vocational education.
7. Developing Sustainable Water management system for rural areas and implementation approaches.
8. Contribution to any national level initiative of Government of India. For eg. Digital India, Skill India, Swatch Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.
9. Spreading public awareness under rural out reach programs. (Minimum 5 programs).
10. Social connectand responsibilities.
11. Plantation and adoption of plants. Know your plants.
12. Organize National integration and social harmony events/workshops/seminars. (Minimum 02 programs).
13. Govt.school Rejuvenation and helping them to achieve good infrastructure.

NOTE:

1. Student/s in individual or in a group Should select any one activity in the beginning of each semester till end of that respective semester for successful completion as per the instructions of NSS officer with the consent of HOD of the department.
2. At the end of every semester, activity report should be submitted for evaluation.

DistributionofActivities-Semesterwisefrom 3rd to 6thsemester

Sem	Topics/ActivitiestobeCovered
3rd Sem for 25 Marks	1. Organic farming, Indian Agriculture (Past, Present, andFuture) Connectivity for marketing. 2. Waste management–Public, Private and Govt organization,5R's. 3. Setting of the information imparting club for women leading to contribution in social and economic issues.
4th Sem for 25 Marks	1. Water conservation techniques–Role of different stake holders–Implementation. 2. Preparing an action able business proposal for enhancing the village income and approach for implementation. 3. Helping local schools to achieve good results and enhance their enrolment in Higher/technical/ vocational education.
5th Sem for 25 Marks	1. Developing Sustainable Water management systems for rural areas and implementation approaches. 2. Contribution to any national-level initiative of the Government of India.Foreg. Digital India, Skill 3. India, Swachh Bharat, Atmanirbhar Bharath, Make In India, Mudra scheme, Skill development programs etc. 4. Spreading public awareness under rural out reach programs. (minimum 5 programs). 5. Social connect and responsibilities.
6th Sem for 25 Marks	1. Plantation and adoption of plants. Know your plants. 2. Organize National integration and social harmony events/workshops/seminars. (Minimum 02 programs). 3. Govt. school Rejuvenation and helping them to achieve good infrastructure.

Course outcomes (CourseSkillSet):

COs	Description
M23BNSK310.1	Understand the importance of his/her responsibilities towards society.
M23BNSK310.2	Analyse the environmental and societal problems/issues and will be able to design solutions for the same.
M23BNSK310.3	Evaluate the existing system and to propose practical solutions for the same for sustain able development.
M23BNSK310.4	Implement government or self-driven projects effectively in the field.
M23BNSK310.5	Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

Pedagogy–Guidelines:

It may differ depending on local resources available for the study as well as environment and climatic differences, location, and time of execution.

Sl. No.	Topic	Group size	Location	Activity execution	Reporting	Evaluation of the Topic
1.	Organic farming, Indian Agriculture (Past, Present, and Future) Connectivity for marketing.	May be individual or team	Farmers land / Villages / roadside / community area /College campus etc	Site Selection / Proper consultation / Continuous monitoring/ Informationboard	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer

2.	Waste management– Public, Private and Govt Organization 5R's.	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Site selection / Proper consultation/ Continuous monitoring/ Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
3.	Setting of the information imparting club for women leading to contribution in social And economic issues.	May be individual or team	Women empowerment groups/ Consulting NGOs&Govt Teams / Collegecampus etc…	Group selection/ proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
4.	Water conservation techniques – Role of differentstakeholders– Implementation.	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemes officers / campusetc…	Siteselection/ Proper consultation/ Continuous monitoring/ Informationboard	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
5.	Preparing an actionable business proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/ Cit y Areas / Grama panchayat/ public associations/ Government Schemesofficers /Campus etc…	Groupselection/ proper consultation / Continuous monitoring / Informationboard	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
6.	Helping local schools to achieve good results and enhance their enrolment in Higher/ technical/ vocational education.	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemes officers / Campus etc…	School selection/ proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
7.	Developing Sustainable Water management system for rural areas and implementation approaches.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /Campus etc…	Site selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer

8.	Contribution to any national-level initiative of the Government of India. For eg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers/ campus etc…	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
9.	Spreading public awareness under rural outreach programs. (Minimum 5 programs). Social connect and responsibilities.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers/ campus etc…	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
10.	Plantation and adoption of plants. Know your plants.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers/ campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
11.	Organize National integration and social harmony events /Workshops /Seminars. (Minimum 02 programs).	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /Campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
12.	Govt. school Rejuvenation and helping them to achieve good infrastructure.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /Campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer

Plan of Action (Execution of Activities For Each Semester)

Sl. No	PracticeSession Description
1.	Lecture session by NSS Officer
2.	Students'PresentationonTopics
3.	Presentation-1, Selection of topic, PHASE-1
4.	Commencement of activity and its progress -PHASE-2
5.	Execution of Activity
6.	Execution of Activity

7.	Execution of Activity
8.	Execution of Activity
9.	Execution of Activity
10.	Casestudy-based Assessment, Individual performance
11.	Sectorwise study and its consolidation
12.	Video-based seminar for10 minutes by each student At the end of the semester with a Report.

- In every semester from 3rd semester to 6th semester, each student should do activities according to the scheme and syllabus.
- At the end of every semester student performance has to be evaluated by the NSS officer for the assigned activity progress and its completion.
- Atlast in 6th semester consolidated report of all activities from 3rd to 6th semester, compiled report should be submitted as per the instructions.

AssessmentDetails:

Weightage	**CIE – 100%**	
Presentation-1 Selection of topic, PHASE-1	**10 Marks**	• Implementation strategies of the project (NSS work). • The last Report should be signed by theNSS Officer, the HOD, and the principal. • Atlast Report shouldbe evaluated by the NSS officer of the institute. • Finally, the consolidated marks sheet should be sent to the university and made available at the LIC visit.
Commencemen t of activity and its progress - PHASE-2	**10 Marks**	
Case Study-based Assessment Individual Performance with Report	**10 Marks**	
Sector-wise study & its consolidation	**10 Marks**	
Video based seminar for 10 minutes by each student Attheend of semester with Report. Activities.	**10 Marks**	
Total marks for the course in each semester	**50 Marks**	

Marks scored for 50 by the students should be Scale down to 25 marks In each semester for CIE entry in the VTU portal.

25 marks CIE entry will be entered in University IA marks portal at the end of each semester 3rd to 6th sem, Report and assessment copy should be made available in the department semester wise

Students should present the progress of the activities as per the schedule in the prescribed practical session in thefield.

There should be positive progress in the vertical order for the benefit of society ingeneral.

Suggested Learning Resources:

Books:

1. NSS Course Manual, Published by NSS Cell, VTU Belagavi.
2. Government of Karnataka, NSS cell, activities reports and manual.
3. Government of India, NSS cell, Activities reports and manual.

3rd Semester	Non Credit Mandatory Course (NCMC) PHYSICAL EDUCATION (SPORTS & ATHLETICS)	M23BPEK310

PHYSICAL EDUCATION (SPORTS& ATHLETICS)

SEMESTER-III

Course Code	**M23BPEK310**	CIE Marks	**100**
Number of Lecture Hours/Week (L: T: P: S)	**(0:0:2:0)**	SEE Marks	
Total Number of Lecture Hours	**-**	Total Marks	**100**
Credits	**0**	Exam Hours	**-**

Course Outcomes: At the end of the course, the student will be able to

M23BPEK310.1: Understand the fundamental concepts and skills of Physical Education, Health, Nutrition and Fitness.

M23BPEK310.2: Familiarization of health-related Exercises, Sports for overall growth and development.

M23BPEK310.3: Create a foundation for the professionals in Physical Education & Sports.

M23BPEK310.4: Participate in the competition at regional/state / national / international levels.

M23BPEK310.5: Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Module-1

Orientation **(5hours)**

- Lifestyle
- Fitness
- Food & Nutrition
- Health &Wellness
- Pre-Fitness test.

Module-2

General Fitness & Components of Fitness: **(15hours)**

- Warming up (Free Hand exercises)
- Strength — Push-up / Pull-ups
- Speed — 30 Mtr Dash
- Agility — Shuttle Run
- Flexibility — Sit and Reach
- Cardiovascular Endurance — Harvard step Test

Module-3

Recreational Activities: **(10hours)**

- Postural deformities.
- Stress management.
- Aerobics.
- Traditional Games.

Scheme and Assessment for auditing the course and Grades:

Sl. No.	Activity	Marks
1.	Participation of student in all the modules	20
2.	Quizzes — 2, each of 15 marks	30
3.	Final presentation/exhibition / Participation in competitions/practical on specific tasks assigned to the students	50
	Total	**100**

3rd Semester	**Non-Credit Mandatory Course (NCMC) YOGA**	**M23BYOK310**

YOGA - SEMESTER-III			
Course Code	**M23BYOK310**		
Number of Lecture Hours/Week (L: T: P: S)	**0:0:2:0**	CIE Marks	**100**
Total Number of Lecture Hours		SEE Marks	**-**
Credits	**0**	Total Marks	**100**
Evaluation Method: Objective type Theory / Practical / Viva- Voice			

Course objectives:

1. To enable the student to have good Health.
2. To practice mental hygiene.
3. To possess emotional stability.
4. To integrate moral values.
5. To attain a higher level of consciousness.

The Health Benefits of Yoga

The benefits of various yoga techniques have been supposed to improve

- body flexibility,
- performance,
- stress reduction,
- attainment of inner peace, and
- self-realization.

The system has been advocated as a complementary treatment to aid the healing of severalailments such as

- coronary heart disease,
- depression,
- anxiety disorders,
- asthma, and extensive rehabilitation for disorders including musculoskeletal problems and traumatic brain injury.

The system has also been suggested as behavioral therapy for smoking cessation and substanceabuse (including alcohol abuse).

If you practice yoga, you may receive these physical, mental, and spiritual benefits:

- **Physical**
  1. Improved body flexibility and balance
  2. Improved cardiovascular endurance (stronger heart)
  3. Improved digestion
  4. Improved abdominal strength
  5. Enhanced overall muscular strength
  6. Relaxation of muscular strains
  7. Weight control
  8. Increased energy levels
  9. Enhanced immune system
- **Mental**
  1. Relief of stress resulting from the control of emotions
  2. Prevention and relief from stress-related disorders
  3. Intellectual enhancement, leading to improved decision-making skills
- **Spiritual**
  1. Life with meaning, purpose, and direction
  2. Inner peace and tranquillity
  3. Contentment

Yoga Syllabus
Semester III
• Yoga, its origin, history and development. Yoga, its meaning, definitions.

- Different schools of yoga, Aim and Objectives of yoga, importance of prayer
- Yogic practices for a common man to promote positive Health
- Rules to be followed during yogic practices by the practitioner
- Yoga its misconceptions,
- Difference between yogic and non-yogic practices
- Surya namaskar prayer and its meaning, Need, importance and benefits of Surya namaskar 12 count, 2rounds
- Asana, Need, importance of Asana. Different types of asanas. Asana its meaning by name, technique,precautionary measures and benefits of each asana
- Different types of Asanas

  a. Sitting
  1. Padmasana
  2. Vajrasana

  b. Standing
  1. Vrikshana
  2. Trikonasana

  c. Prone line
  1. Bhujangasana
  2. Shalabhasana

  d. Supine line
  1. Utthitadvipadasana
  2. Ardhahalasana

Assessment Details (both CIE and SEE)

- Students will be assessed with internal test by a. Multiple choice questions b. Descriptive type questions (Two internal assessment tests with 25 marks/test)
- Final test shall be conducted for whole syllabus for 50 marks.
- Continuous Internal Evaluation shall be for 100 marks (including IA test)

Suggested Learning Resources:

Books:

1. Yogapravesha in Kannada by Ajitkumar
2. Light on Yoga by BKS Iyengar
3. Teaching Methods for Yogic practices by Dr. M L Gharote & Dr. S K Ganguly
4. Yoga Instructor Course hand book published by SVYASA University, Bengaluru
5. Yoga for Children –step by step – by Yamini Muthanna

Web links and Video Lectures (e-Resources):Refer links

1. https://youtu.be/KB-TYlgd1wE
2. https://youtu.be/aa-TG0Wg1Ls

Sl. No.	**Activity**	**Marks**
1.	Participation of student in all the modules	**20**
2.	Quizzes—2,eachof15marks	**30**
3.	Final presentation/exhibition/Participation in competitions/practical on specific tasks Assigned to the students	**50**
	Total	**100**

3rd Semester	Basic Science Course (BS) **DIPLOMA MATHEMATICS-I**	M23BDIPM311

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Calculus	Knowledge of calculus, specifically Successive differentiation-problems.
2.	Basic Concepts Of Complex Numbers	Strong knowledge of Complex Numbers, to solve differential equations.
3.	Basic Mathematics	Knowledge of advanced calculus, and ordinary differential equations Familiarity with identify the dependent and independent variables
4.	BasicConcept of Vector Algebra and Vector Differentiation	Knowledge of basics, solving Mechanical engineering problems using vector calculus.
5.	Previous Coursework	Completionofintroductorycoursesfor ME Stream orarelatedfield

2. Competencies

S/L	Competency	KSADescription
1.	Differential Calculus:	**Knowledge:** Introduction to of basics of Successive differentiation **Skills:** Total derivatives-differentiation of composite functions. Jacobians of order two-Problems Analysis of probabilistic models. **Attitudes:**Applications to mathematical quantities of Partial Differentiation.
2.	Complex Numbers:	**Knowledge:** Basic concept of Complex Numbers. **Skills:** Solving ordinary differential equations arising inengineering applications. **Attitudes:** Appreciation for using ordinary differential equation in Vibration of a rod/membrane.
3.	Vector Differentiation	**Knowledge:** Understanding of basic operations on vector calculus **Skills:** Apply to the heat and mass transfer, oil refinery problems, environmental engineering, **Attitudes:** Appreciation for velocity and acceleration of moving particles, analysis of streamlines.
4.	Integral Calculus	**Knowledge**: Understanding of basic solution of algebraic and transcendental equations: **Skills:** Solve mechanical engineering problems involving Integral Calculus in Engineering analysis. **Attitudes:** solutions to solve mechanical engineering problems involving Integral Calculus.
5.	Ordinary Differential Equations (ODEs):	**Knowledge:** Understanding of basics ordinary differential equations of first order and first degree **Skills:** Solving ordinary and partial differential equations arising inengineering applications, using numerical methods **Attitudes:** Finding approximate solutions to solve mechanical engineering problems.

3. Syllabus

Diploma Mathematics-I SEMESTER–III			
CourseCode	**M23BDIPM311**	CIEMarks	**50**
NumberofLectureHours/Week (L:T:P:S)	**(2:0:0:0)**	SEEMarks	**-**
TotalNumberofLectureHours	**20 Hours**	TotalMarks	**50**

Dean Academics
MIT Mysore

APPROVED
MAHARAJA INSTITUTE OF TECHNOLOGY
MIT
MYSORE
MYSORE
Dean Academics

Principal
MIT Mysore

Credits	**0**	Exam Hours	**00**

Courseobjectives: Thiscoursewillenablestudentsto:

The mandatory learning course M23BDIPM311 viz., Additional Mathematics-I aims to provide basic concepts of complex numbers, vector algebra, differential & integral calculus, vector differentiation and methods of solving first-order differential equations.

Module -1 Differential Calculus: (8 hours)

Successive differentiation-problems. Taylor’s & Maclaurin’s series Expansions-problems. Partial Differentiation: Euler’s theorem (without Proof)-problems on first order derivatives only. Total derivatives-differentiation of composite functions. Jacobians of ordertwo-Problems.

Module -2 Complex Numbers: (8 hours)

Definitions and properties. Modulus and amplitude of a complex number, Argand’s diagram, De-Moivre’s theorem (without proof), Problems. Vector Algebra: Scalar and vectors. Addition, subtraction and multiplication of vectors- Dot andCross products, problems. Scalar triple product, Problems.

Module -3 Vector Differentiation: (8 hours)

Differentiation of vector functions. Velocity and acceleration of a particle moving on a space curve. Scalar and vector point functions. Gradient, Divergence, Curl-simple problems.
Solenoidal and irrotational vector fields-Problems.

Module -4 : Integral Calculus: (8 hours)

Review of elementary integral calculus. Reduction formulae for sinx, cos^{n}x, sin^{n}xcos^{n}x (without proof) and evaluation of these with standard limits-problems.
Double and triple integrals-Simple problems.

Module -5 Ordinary Differential Equations (ODEs): (8 hours)

Introduction-solutions of first order and first-degree differential equations: Variable separable method, Homogeneous differential equations, linear differential equations. Exact differential equations.

Text Books:
1. Higher Engineering Mathematics, B. S. Grewal, Khanna Publishers, 43rd Edition, 2015.
2. Advanced Engineering Mathematics, E. Kreyszig John, Wiley & Sons, 10th Edition, 2015.

Reference Books:
1. Engineering Mathematics, N. P. Bali and Manish Goyal, Laxmi Publishers, 7th Edition, 2007.
2. Higher Engineering Mathematics, H. K. Das and Er. Rajnish Verma, S. Chand & Company

4. **SyllabusTimeline**

S/L	Syllabus Timeline	Description
1	Week 1-2: Differential Calculus:	Successive differentiation-problems. Taylor’s & Maclaurin’s series expansions-problems. Partial Differentiation: Euler’s theorem (without Proof)-problems on first order derivatives only. Total derivatives-differentiation of composite functions. Jacobians of order two-Problems.
2	Week 3-4: Complex Numbers	Definitions and properties. Modulus and amplitude of a complex number, Argand’s diagram, De-Moivre’s theorem (without proof), Problems. Vector Algebra: Scalar and vectors. Addition, subtraction and multiplication of vectors- Dot and Cross products, problems. Scalar triple product, Problems.
3	Week 5-6: Vector Differentiation:	Differentiation of vector functions. Velocity and acceleration of a particle moving on a space curve. Scalar and vector point functions. Gradient, Divergence, Curl-simple problems. Solenoidal and irrotational vector fields-Problems.
4	Week 7-8: Integral Calculus:	Review of elementary integral calculus. Reduction formulae for sin^{n}x, cos^{n}x, sin^{n}xcos^{m}x (without proof) and evaluation of these with standard limits-problems. Double and triple integrals-Simple problems.

5	Week 9-10: Ordinary Differential Equations	Introduction-solutions of first order and first-degree differential equations: Variable separable method, Homogeneous differential equations, linear differential equations. Exact differential equations.
6	Week 11-12: Applications	Applications of the above topics

4. Teaching-LearningProcessStrategies

S/L	TLPStrategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture formatto reinforce Competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance Understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	HigherOrderThinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of Competencies
6	Multiple Representations	Introduce topics in various representation store in force competencies
7	Real-WorldApplication	Discuss practical applications to connect theoretical concepts with real-worldcompetencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to Facilitate deeper understanding of competencies

6. Assessment Details (Only CIE)

The weightage of Continuous Internal Evaluation (CIE) is 100%. The minimum passing mark for theCIE is 40% of the maximum marks (100). A student shall be deemed to have satisfied the academic. Requirements if the student secures not less than 40% (40 Marks out of 100) in the CIE.

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

7. LearningObjectives

S/L	Learning Objectives	Description
1	UnderstandingIntegral calculus and Vector differentiation and its Fundamentals	Students will learn the importance of Integral calculus and Vector differentiation essential for Mechanical engineering.
2	Understanding Fundamentals of ordinary Differential Equations	Students will formulate various mathematical models by using ordinary Differential Equations
3	Proficiency incomplex numbers	Students will become proficient in solving complex numbers problems.
4	Project-BasedLearning	Through hands-on projects, students will apply their knowledge ofMake use of the correlation and regression analysis to fit a suitable mathematical model for the statistical data.

5	CollaborationandCommunication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively,share ideas,and solve problems collectively.
6	Ethical andProfessionalResponsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security,and adhering to Industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

M23BDIPM311.1	**Description**
M23BDIPM311.2	Use derivatives and partial derivatives to calculate the rate of change of multivariate functions.
M23BDIPM311.3	Apply concepts of complex numbers and vector algebra to analyse the problems arising in a related area.
M23BDIPM311.4	Analyse position, velocity and acceleration in two and three dimensions of vector-valued functions.
M23BDIPM311.5	Learn techniques of integration including the evaluation of double and triple integrals.
M23BDIPM311.6	Identify and solve first-order ordinary differential equations.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BDIPM311.1	3													3
M23BDIPM311.2		3											3	
M23BDIPM311.3		3											3	3
M23BDIPM311.4	3												3	
M23BDIPM311.5		3												3
M23BDIPM311.6			3											3
M23BDIPM311	3	3	3										3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**2**	**2**	**2**	**2**	**2**	**10**
Module 2	**2**	**2**	**2**	**2**	**2**	**10**
Module 3	**2**	**2**	**2**	**2**	**2**	**10**
Module 4	**2**	**2**	**2**	**2**	**2**	**10**
Module 5	**2**	**2**	**2**	**2**	**2**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

10. FuturewiththisSubject

The Diploma Mathematics-I course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions ofthis subject extends across various areas, enhancing the students' understanding and skills in thefieldof digital systems. Here aresomenotable contributions:

Calculus: use of calculus quite often in our daily lives. Various fields such as engineering, medicine, biological research, economics, architecture, space science, electronics, statistics, and pharmacology all benefit from the use of calculus.

Complex numbers Imaginary numbers or complex numbers are used in various fields such as: Signal processing: Imaginary numbers can also be applied to signal processing, which is useful in cellular technology and wireless technologies, as well as radar and even biology (brain waves).

4th Semester	**Basic Science Course (BS)** **Mathematics-III for CSE Stream**	**M23BMAT401**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic concepts of Statistic and Curve fitting	Understanding the relationship between one or more predictors and a response variable with the goal of defining a “best fit” model of the relationship Familiarity with fundamental knowledge of algebra course
2.	Calculus	Knowledge of calculus, specifically integration and differentiation and an understanding of complex numbers
3.	Basic Concepts of Linear algebra	Strong knowledge of calculus, linear algebra, complex numbers and trigonometric functio
4.	Basic Mathematics	Knowledge of advanced calculus, linear algebra, and ordinary differential equations Familiarity with identify the dependent and independent variables
5.	Basic Concept of Permutation and Combination	Knowledge of basics set theory, inclusion and exclusion principle (knowing different ways of counting) and calculus (knowing derivatives and integrals)
6.	Previous Coursework	Completion of introductory courses in Basic electronics or a related field

2. Competencies

S/L	Competency	KSA Description
1.	**Statistical Methods**	**Knowledge**: Principle of least squares, Correlations and lines of regressions **Skills**: Apply correlation analysis to build more accurate and efficient models **Attitudes:** Appreciation for the correlation analysis to build more accurate and efficient models
2.	**Fourier Series**	**Knowledge**: Periodic functions, Dirichlet’s condition, Practical harmonic analysis **Skills:** Fourier series to represent periodical physical phenomena in Engineering analysis. **Attitudes:** Appreciation for the role of Fourier series engineering
3.	**Probability Distribution**	**Knowledge:** Understanding of Poisson and Normal Distribution **Skills:** Apply probability for risk assessment in the design of structures such as bridges, dams and buildings **Attitudes:** Appreciation for the role of Probability distribution in risk assessment.
4.	**Sampling Variables**	**Knowledge:** Means**,** Student’s t distribution and Chi square distribution **Skills:** Apply Sampling enables the selection of right data points from within the larger data set to estimate the characteristics of the whole population**.** **Attitudes:** Appreciation for the role of Sampling in industry to check the quality of a product
5.	**ANOVA Technique**	**Knowledge:** Basic Principles of Anova, One-way Anova and two-way Anova **Skills:** Apply ANOVA is used in a business context to help manage budgets by comparing your budget to costs to help manage revenue and inventory **Attitudes:** Appreciation for the role of Anova in random block design
6.	**Significance of means**	**Knowledge:** Understanding the test of significance for two means **Skills:** Apply distribution to test the goodness of fit **Attitudes:** Recognizing the significance of sampling variables and distributions

3. Syllabus

MATHEMATICS-III FOR CSE STREAM **SEMESTER – III**			
Course Code	M23MATS401	CIE Marks	**50**

Number of Lecture Hours/Week (L: T: P: S)	**(2:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Learn to use the Fourier series to represent periodical physical phenomena in engineering analysis
- To enable the students to study Fourier Transforms and concepts of infinite Fourier Sine and Cosine transforms and to learn the method of solving difference equations by the z-transform method.
- To have insight in to Statistical methods, correlation and regression analysis , fitting of curves
- To Provide the principles of statistical inferences and the basics of hypothesis testing with emphasis on some commonly encountered hypotheses.
- To Determine whether an input has a statistically significant effect on the system's response through ANOVA testing.

Module -1 Statistical Methods and Curve Fitting

Curve fitting by the method of least squares, fitting the curve of the forms $y = ax + b$, $y = ax^b$ and $y = ax^2 + bx + c$, Correlation and regression- Karl Pearson's coefficient of correlation and rank correlation, problems. Regression analysis, lines of regression, problems

Module -2 Fourier Series

Introduction to infinite series, convergence and divergence. Periodic functions, Dirichlet's condition. Fourier series of periodic functions with period 2l, Practical harmonic analysis

Module -3 Probability Distribution

Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation, mean and variance. Binomial, Poisson Exponential and Normal Distributions, (Statements only), Problems Joint probability distribution: Joint Probability distribution for two discrete random variables, Expectation, covariance and correlation

Module -4Statistical Inference

Sampling variables, central limit theorem and confidences limit for unknown mean. Test of Significance for means of two small samples, Students't' distribution, Chi-square distribution as a test of goodness of fit. F-Distribution.

Module -5Design of Experiments & ANOVA

Principles of experimentation in design, Analysis of completely randomized design, randomized block design. The ANOVA Technique, Basic Principle of ANOVA, One-way ANOVA, Two-way ANOVA, Latin-square Design

Text Books:

- B.S.Grewal:"HigherEngineeringMathematics",Khannapublishers,44thEd.2018
- E.Kreyszig:"AdvancedEngineeringMathematics",JohnWiley&Sons,10thEd.(Reprint),20

ReferenceBooks

- V.Ramana:"HigherEngineeringMathematics"McGraw-HillEducation,11thEd.
- SrimantaPal&SubodhC.Bhunia:"EngineeringMathematics"OxfordUniversityPress,3rdReprint,2016.
- N.PBaliandManishGoyal:"AtextbookofEngineeringMathematics"LaxmiPublications,Latestedition.
- C.RayWylie,LouisC.Barrett:"AdvancedEngineeringMathematics"McGraw–HillBookCo. Newyork, Latested.
- GuptaC.B,SingS.RandMukeshKumar:"EngineeringMathematicforSemesterIandII",Mc-GrawHill Education (India)Pvt.Ltd2015.
- H.K.DassandEr.RajnishVerma:"Higher Engineering Mathematics "S.Chand Publication(2014).
- JamesStewart:"Calculus"Cengagepublications,7thedition,4thReprint2019.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Statistical Methods and Curve Fitting	Correlation and regression Karl Pearson's coefficient of correlation and rank correlation Regression analysis, lines of regression Fitting the curve of the forms $y = ax + b$ Fitting the curve of the forms $y = \quad ax^b$ Fitting the curve of the forms $y = ax^2 + bx + c$
2	Week 3-4: Fourier Series	Introduction to trigonometric polynomial, trigonometric series. Dirichlet's conditions Fourier Series of periodic functions with period$2l$ Fourier Series of periodic functions with period2π Practical harmonic analysis.
3	Week 5-6: Probability Distribution	Review of basic probability theory. Random variables Problems on Binomial Distribution Problems on Poisson Distribution Problems on Exponential Distribution Problems on Normal Distribution Joint Probability distribution for two discrete random variable
4	Week 7-8: Statistical Inference	Central limit theorem and confidences limit for unknown mean Test of Significance for means of two small samples students 't' distribution, Chi-square distribution as a test of goodness of fit F-Distribution
5	Week 9-10: Design of Experiments & ANOVA	Analysis of completely randomized design Randomized block design The ANOVA Technique, Basic Principle of ANOVA One-way ANOVA, Two-way ANOVA Latin-square Design,
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with programming assignments

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of the concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming	Assign programming tasks to reinforce practical skills associated with

	Assignments	competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject (**duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 questions from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding Fundamentals of Statistics	Students will formulate complete, concise, and correct mathematical proofs. Students will frame problems using multiple mathematical and statistical representations of relevant structures and relationships and solve using standard techniques
2	Proficiency in Fourier series	Students will become proficient in writing a series expansion of even and odd function and also writing the best fitting of the curve using least square method
3	Understanding Sample mean	Student will understand the statistical inference to estimate the uncertainty or sample to sample variation. It allows us to provide a probable range of values for the true values of something in the population
4	Project-Based Learning	Through hands-on projects, students will apply their knowledge of Make use of the ANOVA, or Analysis of Variance, is a test used to determine differences between research results from three or more unrelated samples or groups.
5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23MATS401.1	Apply the concepts of Statistics, Probability, Statistical inference, series and Anova to solve Engineering Problems
M23MATS401.2	Analyze the computer Science Engineering application problems through Least squares, statistical inference and series method
M23MATS401.3	Relate the importance of Test of Significance and ANOVA testing appearing in computer science engineering

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23MATS401.1	**3**												**3**	**3**
M23MATS401.2		**3**											**3**	**3**
M23MATS401.3	**3**												**3**	**3**
M23MATS401	**3**	**3**											**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	**Total**
Module 1	**2**	**5**	**3**	**10**
Module 2	**2**	**5**	**3**	**10**
Module 3	**2**	**5**	**3**	**10**
Module 4	**2**	**5**	**3**	**10**
Module 5	**2**	**5**	**3**	**10**
Total	**10**	**25**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	**Total**
Module 1	**4**	**10**	**6**	**20**
Module 2	**4**	**10**	**6**	**20**
Module 3	**4**	**10**	**6**	**20**
Module 4	**4**	**10**	**6**	**20**
Module 5	**4**	**10**	**6**	**20**
Total	**20**	**50**	**30**	**100**

Conditions for SEE Paper Setting:
Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

- The "Mathematics –III for CSE Stream" course in the third semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:
- **Linear and Nonlinear Regression:** The knowledge gained in this course, regression analysis, curve fitting is the process of specifying the model that provides the best fit to the specific curves in your dataset. Curved relationships between variables are not as straightforward to fit and interpret as linear relationships. For linear relationships, as you increase the independent variable by one unit, the mean of the dependent variable always changes by a specific amount. This relationship holds true regardless of where you are in the observation space.
- **Mathematical Finance:** Probability as a subject in and of itself has rarely been truly appreciated by mathematicians in other disciplines. This has gradually changed over the last 50 years, as occasionally brilliant mathematicians show how it can be used to solve, or to explain, and/or to give intuitive content to thorny mathematical issues. We provide some examples and then give a wild speculation as to where the field, at least in Mathematical Finance, might go in the future
- **Medical research:** In medical research, the ANOVA test can be used to identify the relationship between various types or brands of medications on individuals with migraines or depression. In social sciences, ANOVA tests can be used to study the statistical significance of various study environments on test Scores

- **Statistics:** Statistical inference plays a crucial role in analyzing data and drawing conclusions from statistical analyses, enabling researchers and statisticians to make predictions and decisions based on sample data. By employing methods such as confidence intervals and hypothesis tests, it bridges the gap between sample data and the larger population, facilitating an understanding of underlying patterns and trends. This dynamic field combine’s theory with practical application, essential for professionals in fields ranging from academia to industry, ensuring informed decision-making based on statistical evidence.

4th Semester	**Professional Course (PC)** **ANALYSIS AND DESIGN OF ALGORITHMS**	**M23BCS402**

1. Prerequisites

S/L	Proficiency	Prerequisites
01	**Understanding of Basic Programming Concepts**	Solid understanding of basic programming concepts and data structures. This includes knowledge of variables, control structures (loops, conditionals), arrays, linked lists, stacks, queues, trees, and graphs.
02	**Mathematical Foundations**	A strong foundation in mathematics is essential. This includes knowledge of algebra, calculus, and discrete mathematics. Understanding of topics such as functions, equations, inequalities, sequences, series, and combinatory is crucial for analyzing algorithmic complexity.
03	**Programming Proficiency**	Proficiency in at least one programming language is necessary. Understand concepts such as loops, recursion, functions, and basic algorithmic constructs.
04	**Problem Solving Skills**	Possess strong problem-solving skills. This involves the ability to analyze problems, identify patterns, and devise algorithmic solutions. Prior experience with problem-solving competitions or algorithmic challenges can be beneficial.
05	**Logical Thinking**	The ability to think logically and critically is crucial for understanding complex algorithms and analyzing their performance.
06	**Familiarity with Algorithmic Analysis Techniques(Optional)**	Prior exposure to basic algorithm analysis techniques such as time complexity, space complexity, and Big O notation would be advantageous. This knowledge provides a foundation for understanding more advanced concepts covered in the course.

2. Competencies

S/L	Competency	KSA Description
1	**Foundational Algorithmic Analysis and Design**	**Knowledge:** Understanding the fundamentals of the analysis of algorithm efficiency and learn toconduct mathematical analysis for both non-recursive and recursive algorithms**.** Understanding basic algorithmic strategies and their applications to simple problem-solving scenarios. **Skills:** Able to evaluate algorithm performance using analysis frameworks, asymptotic notations, and efficiency classes. **Attitudes:** Critical and analytical attitude and adopt a thorough and detail-oriented mindset to analyze more complex scenarios.
2	**Algorithmic Strategy and Problem-Solving**	**Knowledge:** Gain insights into solving complex problems through comprehensive, albeit computationally expensive, methods. Understanding how to reduce problem size to achieve solutions efficiently. Imparting knowledge on breaking down problems into smaller, more manageable sub problems, solving them independently, and combining their solutions. **Skills:** Develop skills in systematically exploring all possible solutions to find the optimal one. Ability to simplify complex problems by reducing their size. Proficiency in breaking down problems into smaller sub problems, solving them independently, and efficiently combining the results. **Attitudes:** Analytical thinking and adaptability, step-by-step problem-solving mentality
3	**Advanced Data**	**Knowledge:**

	Structures and Algorithm Optimization	Understanding of advanced data management and optimization. Learn to balance and optimize the use of computational resources. **Skills:** Efficient organization and manipulation of data structures to optimize algorithm performance. Ability to analyze and optimize algorithms by making strategic tradeoffs between memory usages and processing speed. **Attitudes:** Precisionandsystematic thinking forcareful data structure transformations and efficient management of hierarchical data. **S**trategic thinkingandoptimization skills, learning to balance competing resources.
4	**Algorithmic Problem-Solving and Optimization**	**Knowledge:** Understanding the construction of optimal solutions by making locally optimal choices at each step, enabling efficient problem-solving in various domains. **Skills:** Proficiency in applying algorithms like Warshall's and Floyd's for graph-related problems. Ability to make locally optimal choices at each step to construct globally optimal solutions, enhancing their problem-solving skills across various domains, including graph theory and data compression. **Attitudes:** Problem-solving skillsand ability to optimize solutions by breaking down complex problems into smaller sub problems. Strategic decision-makingandefficiency.
5	**Advanced Problem-Solving and Algorithmic Analysis**	**Knowledge:** Understanding the theoretical boundaries of algorithmic efficiency and the inherent difficulty of certain types of computational problems**.** **Skills:** Critical thinking skills to understand the theoretical boundaries of computational problem-solving, as well as the ability to analyze the complexity of problems and recognize NP-Complete scenarios. Acquire problem-solving skills such as systematic exploration and decision-making. **Attitudes:** Perseverance and adaptability, to tackle challenging problems through iterative exploration, strategic optimization.

3. Syllabus

Analysis & Design of Algorithms
SEMESTER – IV

Course Code	**M23BCS402**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

. To learn and apply methods for analyzing algorithms and evaluating their performance.

:. To demonstrate the efficiency of algorithms using various asymptotic notations.

. To employ a range of algorithm design techniques, including brute force, greedy algorithms, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, backtracking, and branch-and-bound to solve complex problems.

. To gain a solid understanding of P and NP complexity classes and their implications in algorithm design.

Module -1

INTRODUCTION: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving.

FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms.

BRUTE FORCE APPROACHES: Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching.

Chapter 1 (Sections 1.1,1.2), Chapter 2(Sections 2.1,2.2,2.3,2.4), Chapter 3(Section3.1,3.2)

Module -2

BRUTE FORCE APPROACHES (contd..): Exhaustive Search (Travelling Salesman problem and Knapsack Problem).

DECREASE-AND-CONQUER: Insertion Sort, Topological Sorting.

DIVIDE AND CONQUER: Merge Sort, Quick Sort, Binary Tree Traversals, Multiplication of Large Integers and Strassen's Matrix Multiplication.

Chapter 3(Section 3.4), Chapter 4 (Sections 4.1,4.2), Chapter 5 (Section 5.1,5.2,5.3, 5.4)

Module -3

TRANSFORM-AND-CONQUER: Balanced Search Trees, Heaps and Heapsort.

SPACE-TIME TRADEOFFS: Sorting by Counting: Comparison counting sort, Input Enhancement in String Matching: Horspool's Algorithm.

Chapter 6 (Sections 6.3,6.4), Chapter 7 (Sections 7.1,7.2)

Module -4

DYNAMIC PROGRAMMING: Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.

THE GREEDY METHOD: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes.

Chapter 8 (Sections 8.1,8.2,8.4), Chapter 9 (Sections 9.1,9.2,9.3,9.4)

Module -5

LIMITATIONS OF ALGORITHMIC POWER: Decision Trees, P, NP, and NP-Complete Problems. **COPING WITH LIMITATIONS OF ALGORITHMIC POWER**: Backtracking (n-Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NP-Hard problems (Knapsack problem).

Chapter 11 (Section 11.2, 11.3), Chapter 12 (Sections 12.1,12.2,12.3)

Textbooks

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.
2. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Press.

Reference books

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education) Design and Analysis of Algorithms, Dave and Dave, Pearson.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: **INTRODUCTION AND BRUTE FORCE APPROACHES**	**Knowledge:** Comprehend the core principles of algorithm efficiency analysis and acquire the ability to perform mathematical evaluations for both non-recursive and recursive algorithms. Gain insight into fundamental algorithmic strategies and their practical applications to basic problem-solving scenarios. **Skills:** Capable of assessing algorithm performance utilizing analytical frameworks, asymptotic notations, and efficiency classes.
2	Week 3-5: **DECREASE-AND-CONQUER AND DIVIDE AND CONQUER APPROACHES**	**Knowledge:** Acquire the ability to tackle complex problems using thorough, though computationally intensive, methods. Learn techniques for reducing problem size to achieve efficient solutions. **Skills:** Cultivate the ability to systematically explore all potential solutions to identify the optimal one. Enhance proficiency in simplifying complex problems by reducing their size. Develop skills in decomposing

		problems into smaller sub problems, solving them independently, and efficiently integrating their solutions.
3	**Week 6-8:** **TRANSFORM-AND-CONQUER APPROACHES AND SPACE-TIME TRADEOFFS**	**Knowledge:** Develop an understanding of advanced data management and optimization techniques. Acquire the ability to balance and optimize the utilization of computational resources. **Skills:** Efficiently organize and manipulate data structures to enhance algorithm performance. Develop the ability to analyze and optimize algorithms by strategically balancing memory usage and processing speed.
4	Week 9-11: **DYNAMIC PROGRAMMING AND THE GREEDY METHOD**	**Knowledge:** Develop an understanding of constructing optimal solutions by making locally optimal choices at each step, thereby enabling efficient problem-solving across various domains. **Skills:** Gain proficiency in applying algorithms such as Warshall's and Floyd's for solving graph-related problems. Develop the ability to make locally optimal choices at each step to construct globally optimal solutions, thereby enhancing problem-solving skills across various domains, including graph theory and data compression.
5	Week 11-13: **LIMITATIONS OF ALGORITHMIC POWER**	**Knowledge:** Develop a comprehensive understanding of the theoretical limits of algorithmic efficiency and the intrinsic complexity associated with various computational problems. **Skills:** Acquire critical thinking skills to grasp the theoretical limits of computational problem-solving, alongside the capability to analyze problem complexity and identify scenarios classified as NP-Complete.
6	Week 13-15: **COPING WITH LIMITATIONS OF ALGORITHMIC POWER**	**Knowledge:** Study the principles of computational complexity theory, including classes such as P, NP, and NP-complete, and recognizing the inherent challenges in solving problems that fall within these categories. By delving into these theoretical boundaries, one gains insight into why certain problems are computationally hard to solve and the implications this has for algorithm design and practical problem-solving. **Skills:** Develop problem-solving abilities encompassing systematic exploration and strategic decision-making.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.

8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50)in the semester-end examination(SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester End Examination:

Theory SEE will be conducted as per the scheduled timetable, with common question papers for the subject (**duration 03 hours)**

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Algorithm Fundamentals**	Students will comprehend the definition and significance of algorithms in problem-solving.
2	**Analyzing Algorithm Efficiency**	Students will apply a structured framework to assess algorithmic performance in terms of time and space complexity.
3	**Mastering Brute Force Approaches**	Students will gain proficiency in implementing and evaluating brute force algorithms like Selection Sort, Bubble Sort, and Sequential Search.
4	**Exploring Various Algorithmic Strategies**	Students will explore and apply strategies like Merge Sort, Insertion Sort, Balanced Search Trees, and Heapsort.
5	**Advanced Algorithmic Techniques**	Students will gain expertise in implementing Dynamic Programming algorithms for problems like the Knapsack Problem, and Greedy Method for tasks like Prim's Algorithm and Huffman Trees
6	**Understanding Algorithmic Limitations and Coping Strategies**	Students will identify limitations of algorithms including NP-Completeness, and apply coping strategies like Backtracking, Branch-and-Bound, and Approximation Algorithms

7	**Synthesizing Knowledge and Applying in Problem Domains**	Students will synthesize their understanding of various algorithmic techniques to tackle complex problems in real-world scenarios, demonstrating proficiency in algorithm design and analysis.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS402.1	Apply asymptotic notational methods to analyze the performance of algorithms in terms of time complexity.
M23BCS402.2	Demonstrate divide-and-conquer approaches, decrease-and-conquer approaches and transform & conquer and dynamic programming design approaches to solve computational problems.
M23BCS402.3	Design dynamic programming algorithms for optimization problems greedy algorithms for graph problems. Evaluate greedy algorithms for graph problems.
M23BCS402.4	Evaluate problem instances in terms of P, NP, and NP-Complete classes.
M23BCS402.5	Create backtracking algorithms, branch & bound and approximation methods.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS402.1	**3**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**
M23BCS402.2	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BCS402.3	**-**	**-**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BCS402.4	**3**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BCS402.5	**3**	**-**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BCS402	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Analysis and Design of Algorithms" course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

- **Software Development and Engineering**

  Students can become software developers, systems architects, or software engineers, roles that are in high demand across industries like technology, finance, healthcare, and more. Companies such as Google, Microsoft, and Apple highly value these skills.
- **Data Science and Big Data Analytics**

  Careers as data scientists, data analysts, and big data engineers are promising. These professionals are sought after in sectors such as e-commerce, healthcare, finance, and social media, where data-driven decision-making is pivotal.
- **Artificial Intelligence (AI) and Machine Learning (ML)**

  Opportunities include roles as AI/ML engineers, data scientists, and research scientists. This field is rapidly growing and is fundamental to advancements in automation, robotics, and predictive analytics.
- **Cyber security**

  Students can pursue careers as cybersecurity analysts, cryptographers, or information security managers. These roles are critical in protecting digital assets and information systems from threats.
- **High-Performance Computing (HPC)**

  Careers as HPC specialists, computational scientists, and engineers are available in fields like climate modelling, bioinformatics, and physics, where solving large-scale problems is essential.
- **Academic Research and Teaching**

  Students can become researchers or professors, contributing to academia by teaching and exploring new areas in algorithm design and analysis. This path is essential for the ongoing development of computer science as a discipline.

4th Semester	Professional Course (PC) MICROCONTROLLER	M23BCS403

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Understanding of Electronics	Basic knowledge of electronics and digital circuits is important to grasp the fundamentals of how microcontrollers work and how they interact with other components.
2.	Boolean Logic	We often don't deal directly with the ANDs and ORs of the low-level functionality, but it's worth knowing the difference.
3.	Basic Programming Skills	We should have a good understanding of programming fundamentals. Experience with languages like C or C++ is particularly useful as these are commonly used for microcontroller programming.
4.	The basics of the assembly language for the Microcontroller	We rarely have to work at the assembly level nowadays, but it's useful to be able to interpret what's going on in the debugger.
5.	Knowledge of Microcontroller Architecture	Understanding the architecture of the specific microcontroller you are working with is essential. This includes knowledge of memory organization, input/output ports, timers, interrupts, and other core features.
6.	Ability to Read Datasheets	Microcontroller datasheets contain detailed information about the microcontroller's functionality, pinouts, registers, and more. Being able to read and understand datasheets is crucial for working effectively with a microcontroller.
7.	Problem-Solving Skills	Microcontroller programming often involves debugging and troubleshooting complex issues. Strong problem-solving skills are essential for identifying and resolving issues in your code.

2. Competencies

S/L	Competency	KSA Description
1.	**Understanding 8051 Architecture**	**Knowledge:** Solid **understanding of 8051 architecture:** Students will gain a thorough grasp of the 8051's internal structure, including its CPU, memory organization (internal RAM and ROM), special function registers, and addressing modes. Familiarity **with the 8051 instruction set:** The course will equip to understand and utilize the core set of instructions the 8051 can execute. This allows you to write assembly language programs that control the microcontroller's behavior. **Skills:** Ability to understand and program I/O Ports, Counters and Timers. **Attitudes:** Appreciation for the importance assembly level language
2.	**Assembly Language Programming**	**Knowledge:** Students will become proficient in programming the 8051 microcontroller using assembly language, enabling them to write efficient code to control hardware peripherals and implement various functionalities. **Skills:** Analyzing to interface peripherals, handle interrupts, and execute tasks efficiently. **Attitudes:** Appreciation for the importance assembly level language and C.
3.	**Peripheral**	**Knowledge:**

	Interfacing	Students will learn how to interface the 8051 microcontroller with external devices such as sensors, actuators, displays, and communication modules. They will gain practical experience in designing circuits and writing code to interact with these peripherals. **Skills:** Master interfacing various peripherals. **Attitudes:** Appreciation of serial data communication
4.	**Interrupt Handling**	**Knowledge:** Understanding interrupt concepts-how to program the interrupts. **Skills:** Developing skills to manage and prioritize interrupts effectively for timely response in embedded applications. **Attitudes:** Openness to learning and using hardware description languages for design.
5.	**Project-based Learning**	**Knowledge:** Students apply their knowledge of the 8051 microcontroller to design and implement embedded systems projects. Through these projects, students will demonstrate their ability to integrate theory with practice and solve real-world problems. **Skills:** Apply acquired knowledge to design and develop projects ranging from simple LED blinking to advanced automation systems, **Attitudes:** By fostering these positive attitudes toward project-based learning, educators can create an environment where students are motivated, engaged, and empowered to develop their skills in microcontroller programming effectively.
6.	**Documentation and Reporting**	**Knowledge:** By acquiring knowledge in these areas, students can effectively document their microcontroller programming projects, communicate their findings, and collaborate with others in the field. **Skills:** Developing these skills will enable students to effectively document their microcontroller programming projects, communicate their findings, and collaborate with others in the field. **Attitudes:** By fostering these positive attitudes toward documentation and reporting, microcontroller programmers can contribute to the success of their projects by ensuring that information is effectively communicated, stakeholders are well-informed, and project outcomes are well-documented for future reference and improvement.
7.	**Understanding Industry Standards**	**Knowledge:** By understanding industry standards in microcontroller programming, developers can design, develop, and deploy embedded systems that meet quality, safety, security, and regulatory requirements, while also fostering interoperability and compatibility with other systems and devices. **Skills:** Ability to research and identify relevant industry standards, protocols, and regulations applicable to microcontroller programming projects. This includes accessing and interpreting documentation from standards organizations, regulatory bodies, and industry publications. **Attitudes:** By fostering positive attitudes towards understanding industry standards, microcontroller programmers can contribute to the success of their projects by ensuring compliance, interoperability, and quality in their work.

3. Syllabus

Microcontroller
SEMESTER – IV

Course Code	**M23BCS403**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- To describe architecture and operation of Microcontroller 8051 and ARM microcontrollers.
- To familiarize the students with the programming and interfacing of andmicrocontrollers.
- To familiarize the students with cache memory management in ARM microcontrollers.
- To provide strong foundation for designing real world applications using microprocessors andmicrocontrollers.

Module -1

8051 Microcontroller: Microprocessor and Microcontrollers, Four-Bit to Thirty-two-Bit Microcontrollers, 8051 Microcontroller Hardware, I/O Pins, Ports and Circuits.
Text 1: 1.1, 1.3, 3.1, 3.2

Module -2

External Memory, Counters and Timers, Addressing Modes, External Data Moves, Logical Operations –Byte and Bit Level, Rotate and Swap Operations, Arithmetic Operations-Flags, Incrementing and Decrementing, Addition, Subtraction, Multiplication and Division.
Text 1: 3.3, 3.4, 5.1, 5.2, Chapter 6, Chapter 7

Module -3

Jump and Call Instructions-jump and call range, jumps, Calls and Subroutines, Interrupts and Returns, More Details on Interrupts, Sample Programs. Serial Data Communication
Text 1: Chapter 8, Chapter 11 (11.2 Only)

Module -4

Introduction to ARM Processor- RISC Design Philosophy, ARM Design Philosophy-Instruction Set for Embedded System, Embedded System Hardware-ARM BUS Technology, AMBA Bus Protocol, Memory, Peripherals. Embedded System Software-Initialization Code, Operating System, Application, Registers, CPSR
Text 2: Chapter 1, Chapter 2: 2.1, 2.2

Module -5

ARM Processor Fundamentals- Pipeline- Characteristics, Exceptions, Interrupts and Vector table. Caches-Memory Hierarchy and Cache memory, Cache architecture, Policy Coprocessor 15 and Caches, Flushing and Cleaning, Lockdown, Caches and Software Memory. Advanced DSP and SIMD Support in ARMv6.
Text 2: Chapter 2 (2.3 to 2.5), Chapter 12, Chapter 15.1

Exercise Programs (For Demonstration Purpose)

Using suitable simulation software, demonstrate the operation of the following Programs:

1	ALP to add and Subtract 10 bytes of data.
2	ALP to multiply and divide two numbers.
3	ALP to transfer 10 bytes of data from external RAM location starting with 2000h to internal RAM starting from 30h.
4	ALP to count the number of even numbers and number of odd numbers in an array of ‘N’ bytes of data.
5	ALP to count the number of +ve numbers and number of -ve numbers in an array of ‘N’ bytes of data.
6	ALP to find whether the given number is prime or not.
7	ALP to find the factorial of a given number.
8	ALP to check whether the given byte of data is palindrome.
9	ALP to find the square root of a number.
10	ALP to sort an array of numbers in ascending or descending order.

Text Books:

1. “The 8051 Microcontroller” –Kenneth Ayala- Third Edition.
2. ARM System Developer’s Guide-Designing and Optimizing System Software- Andrew N Sloss, Dominic Symes, Chris Wright.

Reference Books:

1. “The 8051 Microntorller and Embedded Systems using Assembly and C” –Muhammad Ali Mazidi, Janice GillispieMazidi, Rolin D Mcinalay- Second Edition.
2. “Microcontrollers-Architechture, Programming, Interfacing and System Design”-Raj Kamal

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Module 1: 8051 Microcontroller	**Competency**: **Architectural Understanding** **Knowledge:** Difference between microprocessor and microcontroller. Understand the architectural features and PIN out of 8051 **Skills:** Understanding the design concept of Microcontroller
2	Week 3-4: Module 2: Addressing mode and Instruction Sets	**Competency**: **Programming Techniques** **Knowledge:** Impart knowledge about assembly language programs of 8051. **Skills:** Learn programming instructions and addressing modes.
3	Week 5-6: Module 3: Instruction Sets	**Competency**: **Programming Techniques and serial data communication** **Knowledge:** Impart knowledge about assembly language programs of 8051. **Skills:** Learn programming instructions with programming.
4	Week 7-8: Introduction to ARM	**Competency:** Architectural Understanding of ARM Microcontroller **Knowledge:** knowledge of architecture of ARM 7processor-LPC2148 **Skills:** Understand the THUMB state and achieving competency in assemblyprogramming of ARM.
5	Week 9-10: ARM Processor Fundamentals	**Competency:** Architectural Understanding of ARM Microcontroller **Knowledge:** Pipeline concepts. Exceptions, Interrupts and Vector Table **Skills:** Importance and characteristics of pipeline.
6	Week 11-12: Caches, Advanced DSP and SIMD Support in ARMv6	**Competency:** Memory management **Knowledge:** Understanding memory hierarchy, Cache architecture. **Skills:** Flushing and cleaning, Lockdown of Caches,

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of Verilog concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order	Pose HOTS questions to stimulate critical thinking related to each competency.

	Thinking (HOTS) Questions:	
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50)in the semester-end examination(SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the Architecture of 8051 Microcontroller	Students will Outline the 8051 architecture, registers, internal memory organization, addressing modes. Discuss 8051 addressing modes, instruction set of 8051, accessing data and I/O port programming.
2	Developing assembly level programs	Students will learn to design and implement 8051 programs for time delay, I/O operations, I/O bit manipulation, logic and arithmetic operations, data conversion and timer/counter programming.
3	Project-Based Learning	Through exercise Programs, students will apply their knowledge of assembly language program todesign various real time projects.
4	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
5	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS403.1	Describe the architecture, register and memory organization, interrupt structure, of the 8051 microcontrollers.
M23BCS403.2	Outline the design philosophy of ARM and RISC Controllers, BUS technology and protocols used to connect different peripherals.
M23BCS403.3	Develop programs in assembly language, using appropriate addressing modes with suitable 8051 instruction sets.
M23BCS403.4	Identify the hierarchy, architecture, policies, of Cache memory.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS403.1	3	-	-	-	-	-	-	-	2	-	-	-	-	2
M23BCS403.2	3	2	2	-	-	-	-	-	-	-	-	-	-	2
M23BCS403.3	3	2	2	-	-	-	-	-	-	-	-	-	-	2
M23BCS403.4	3	3	-	-	-	-	-	-	-	-	-	-	-	2
M23BCS403	3	2.6	2											2

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**10**				**10**
Module 3			**10**			**10**
Module 4				**10**		**10**
Module 5					**10**	**10**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**20**				**20**
Module 3			**20**			**20**
Module 4				**20**		**20**
Module 5					**20**	**20**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Microcontroller and Introduction to ARM" course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems, assembly language and embedded C Programming. Here are some notable contributions:

- These courses cover topics including hardware design ideas, Application Special Instruction Programmers, Application Specific Integrated Circuits.

- **Embedded Systems:** Understanding 8051 and ARM microcontrollers and its application in digital system design is crucial for students pursuing courses related to embedded systems. The ability to model, simulate, and synthesize digital systems using assembly and embedded C is directly applicable in the design and implementation of embedded systems.
- **Project Work and Research:** The hands-on experience gained through programming assignments, problem-solving, and project work in design using assembly language and embedded C prepares students for more extensive projects in their later years. It equips them with the skills needed for research in the field of digital systems.
- **Industry Applications:** The course provides practical skills that are directly applicable in industries related to automotive, industrial, consumer, communication, medical and other Applications Graduates are well-prepared to contribute to industries developing digital hardware and systems

  In summary, the "Microcontroller and Introduction to ARM" course serves as a stepping stone, equipping students with foundational knowledge and skills that are essential for the subsequent courses in their B.E program and for their future careers in various technology-related fields.

4th Semester	**Integrated Professional Course (IPC)** **DATABASE MANAGEMENT SYSTEMS**	**M23BCS404**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Computer Literacy	A solid understanding of how computers work, file management, and using software applications is essential.
2	Fundamentals of Data and Information	Familiarize yourself with the concepts of data, information, and knowledge. Understand the differences between structured and unstructured data.
3	Basic Programming Concepts	While not mandatory, a familiarity with programming concepts can be helpful, especially if you intend to work with databases in a software development context.
4	Operating System Concepts	Familiarity with concepts like file systems, memory management, and process scheduling can help you understand how a DBMS interacts with the underlying operating system.
5	Problem-Solving Skills	Develop your analytical and problem-solving skills, as designing efficient and effective databases often requires making trade-offs and optimizing for different scenarios.

2. Competencies

S/L	Competency	KSA Description
1	**Data Modeling**	**Knowledge:** Understand the principles of data modeling. **Skills:** Entity-Relationship diagrams (ERDs), **Attitudes:** These concepts help design efficient and organized database.
2	**Relational Algebra and Set Theory**	**Knowledge:** Gain basic knowledge of relational algebra and set theory. **Skills:** The knowledge used to interact with relational databases. **Attitudes:** The foundation of relational databases.
3	**SQL (Structured Query Language):**	**Knowledge:** the basics of SQL, the standard language for data query. **Skills:** Writing queries to retrieve, update, and manipulate data. **Attitudes:** Acquired skill to be used for querying with relational databases.
4	**Normalization**	**Knowledge:** Learn about database normalization. **Skills:** To eliminate redundancy and improve data integrity. **Attitudes:** Understand the concept of normalization for optimizing query performance.
5	**Database applications**	**Knowledge:** Gain insight into query optimization strategies. **Skills:** To design data base structure for a particular application. **Attitudes:** To enhance database performance.

3. Syllabus

Database Management System
SEMESTER – IV

Course Code	**M23BCS404**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	40 Hours Theory + 20 Hours Practical	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course objectives:

- To provide a strong foundation in database concepts, technology, and practice.
- To practice SQL programming through a variety of database problems.
- To understand the relational database design principles.
- To demonstrate the use of concurrency and transactions in database.

- To design and build database application for real world problems.
- To become familiar with database storage structures and access techniques.

Module -1

Introduction to DBMS and Database Design **8 hour**

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.

Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces. The Database System environment.

Conceptual Data Modeling using Entities and Relationships: Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization.

Text-1: CH-1.1-1.8, 2.1-2.6, 3.1-310

Module -2

Relational Models **8hour**

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.

Relational Algebra and Calculus: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra, Tuple relational calculus, Domain relational calculus. Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping.

Text-1 : CH-5.1-5.3,8.1-8.7,9.1

Module -3

SQL **8 hour**

SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL : Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.

Text-1 CH-6.1-6.5,7.1-7.4

Module -4

Normalization: **8 hour**

Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Text-1: CH-14.1-14.7

Module -5

Database Application Development: **8 hour**

Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures.

Case study: The internet Bookshop.

Internet applications: The three tier application architecture.

Text-2 : CH-6.1-6.6, 7.5

PRACTICAL COMPONENT

1 Consider the schema for College Database:

STUDENT(USN, SName, Address, Phone, Gender)
SEMSEC(SSID, Sem, Sec)
CLASS(USN, SSID)
COURSE(Subcode, Title, Sem, Credits)
IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to :

1. List all the student details studying in fourth semester 'C' section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '4MH22CS200' in all Courses.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:

	If FinalIA = 17 to 20 then CAT = ‘Outstanding’ *If FinalIA = 12 to 16 then CAT = ‘Average’* *If FinalIA< 12 then CAT = ‘Weak’* Give these details only for 4th semester ALL section students.
2	Consider the schema for Movie Database: **ACTOR(Act_id, Act_Name, Act_Gender)** **DIRECTOR(Dir_id, Dir_Name, Dir_Phone)** **MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)** **MOVIE_CAST(Act_id, Mov_id, Role)** **RATING(Mov_id, Rev_Stars)** Write SQL queries to 1. List the titles of all movies directed by ‘Mani Rathnam’. 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and also in a movie after 2024(use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed by ‘Shankar’ to 5.
3	Consider the schema for Bus ticket reservation Database: **PASSENGER (P_id, P_Name, P_Gender,P_city)** **AGENCY(A_id,A_Name,A_city)** **BUS(B_id,B_date, B_time, B_scr, B_dest)** **BOOKING(P_id, A_id,B_id,B_date,B_time)** Write SQL queries to 1. Get the Complete Details of all the Buses to MIT-Mysore. 2. Find only the Bus Number for Passenger with PID 123 for buses to Mandya, before 05/05/2024. 3. Find the Passenger Name for those who don’t have any booking in any buses. 4. Get the Details of the buses that are scheduled on both dates 01/02/2024 and 02/02/2024 at 16:00 hrs. 5. Find the details of male passengers who are associated with “Happy Smiling” Agency.
4	Consider the schema for Employee salary Database: **EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo)** **DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate)** **DLOCATION(DNo,DLoc)** **PROJECT(PNo, PName, PLocation, DNo)** **WORKS_ON(SSN, PNo, Hours)** Write SQL queries to 1. Retrieve the employee numbers of all employees who work on project located in Mysore, Hassan, or Mangalore 2. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive) 3. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with all the details of Employee. 4. Select the names of employees whose salary is greater than the average salary of all employees in department 10. 5. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.
5	Consider the schema for Matrimonial Database : **ENROLL(E_Name,E_Gender,E_Age,E_Qualification, E_Salary,E_Address, E_ City)** **WORKS(E_name, E_salary, E_city)** **MIRRAGE_BUREAU(MB_name,MB_city,MB_charge)** Write SQL queries to : 1. Find the Names and Cities for all the Groom, who work for MIT-Mysore and earn more than Rs.60,000/- as salary. *2.* Find the Company that has the Least Fee for Marriage Service. 3. Find the name of all the Brides in the database who live in the same cities and on the same

	street as do their Groom. 4. Find the names of Groom in the database, whose qualification and age is same as bride. 5. Fine the name of the Groom in the database, who earns more than all Bride lives in “Mandya”.
Demonstration Experiments (For CIE only – not to be included for SEE)	
1	Hospital Database Management system.
2	Timetable allotment and scheduling system.
3	E-commerce database management system
Text Books:	
1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson. 2. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw-Hill, 3rd Edition.	
Reference Books:	
1. Data base System Concepts, Silberschatz, Korth, McGraw hill, Sixth Edition **2.** An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson, Eight Edition	

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3	**Knowledge:** Understand the principles of data modeling **Skills:** Entity-Relationship diagrams (ERDs). These concepts help design efficient and organized database.
2	Week 4-6	**Knowledge:** Gain basic knowledge of relational algebra and set theory. **Skills:** The knowledge used to interact with relational databases and the foundation of relational databases.
3	Week 7-9	**Knowledge:** The basics of SQL, the standard language for data query. **Skills:** Writing queries to retrieve, update, and manipulate data.
4	Week 10-12	**Knowledge:** Learn about database normalization to eliminate redundancy and improve data integrity. **Skills:** Understand the concept of normalization for optimizing query performance.
5	Week 13-15	**Knowledge:** Gain sight into query optimization strategies to enhance database performance. **Skills:** To design data base structure for a particular application.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Using traditional lecture methods and ICT as and when needed.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance learning.
3	Collaborative Learning	Encourage collaborative learning approaches for peer learning.
4	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application.
5	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
6	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50)in the semester-end examination(SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Theory Course with 4 credits: Integrated Professional Core Course (IPC)

Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Theory (A)	Internal Assessment-Tests (A)	2	**60%**	**15**	**06**
	Assignments/Quiz/Activity (B)	2	**40%**	**10**	**04**
	TotalMarks		**100%**	**25**	**10**
Components		**Number**	**Weightage**	**Max. Marks**	**Min. Marks**
Laboratory(B)	Record Writing	Continuous	**60%**	15	06
	Test at the end of the semester	1	**40%**	10	04
	TotalMarks		**100%**	**25**	**10**

FinalCIE Marks =(A) + (B)

Semester End Examination pattern:

Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.

1. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
2. The students have to answer 5 full questions selecting one full question from each module.
3. The question paper may include at least one question from the laboratory component.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives
1	To provide a strong foundation in database concepts, technology, and practice.
2	To practice SQL programming through a variety of database problems.
3	To understand the relational database design principles.
4	To design and build database application for real world problems.
5	To become familiar with database storage structures and access techniques.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs):

COs	Description
M23BCS404.1	Understand and apply the basic elements of a relational database management system.
M23BCS404.2	Apply various constraints, techniques and Structured Query Language (SQL) statement for database operations.
M23BCS404.3	Analyze various database models and normalization for the given application.
M23BCS404.4	Design and develop entity relationship model and database application.

CO-PO-PSO Mapping:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS404.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS404.2	3	-	-	-	2	-	-	-	-	-	-	-	3	3
M23BCS404.3	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCS404.4	-	-	3	-	-	-	-	-	-	-	-	-	-	3
M23BCS404	**3**	**3**	**3**		**2**								**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**				**-**	**10**
Module 2		**10**			**-**	**10**
Module 3			**10**		**-**	**10**
Module 4				**10**	**-**	**10**
Module 5	**3**	**2**	**3**	**2**	**-**	**10**
Total	**13**	**12**	**13**	**12**	**-**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**				**-**	**20**
Module 2		**20**			**-**	**20**
Module 3			**20**		**-**	**20**
Module 4				**20**	**-**	**20**
Module 5	**5**	**5**	**2**	**5**	**-**	**20**
Total	**25**	**25**	**25**	**25**		**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10.Future with this Subject:

- Data Organization and Storage: Companies can store their data in databases in a structured, organized manner, making it simpler to access and analyze.
- Data Analysis: Databases contain a lot of data, and with the correct tools, organizations can analyze that data to find insights that will help them make business decisions and strategies.
- Efficiency: Databases give companies a centralized area to keep their data, making it more straightforward for staff to retrieve the data they want, minimizing duplication of work and boosting efficiency.
- Security & Privacy: Databases let companies control who has access to their data, ensuring that only authorized users may see and change it. This aids in preventing unauthorized access to and breaches of vital consumer and corporate information.
- This course is the foundation for many other courses to follow such as cloud storage, distributed data storage, block chain, Big data, Quantum computing etc.,

4thSemester	**Professional Course (PC) INTRODUCTION TO ARTIFICIAL INTELLIGENCE**	**M23BCS405**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Programming Skills	While not always mandatory, having a foundational understanding of programming concepts can be beneficial. Python is widely used in AI due to its simplicity and robust libraries for machine learning and data manipulation.
2	Mathematics Fundamentals	Familiarity with basic mathematics concepts such as algebra, calculus, probability, and statistics is essential for understanding the algorithms and models used in AI.
3	Understanding of Data	Knowledge of how data is collected, structured, and processed is crucial in AI. This includes familiarity with databases, data formats, and data preprocessing techniques.
4	Curiosity and Critical Thinking	AI involves problem-solving and continuous learning. Having a curious mindset and the ability to think critically about different AI applications and their implications is important.
5	Books and Research Papers	Reading introductory books and research papers on AI can provide a deeper understanding of fundamental concepts and current trends in the field.
6	Hands-on Projects	Building AI projects, even simple ones, is invaluable for gaining practical experience and reinforcing theoretical concepts. There are many resources and datasets available online for practicing AI projects.
7	Community Engagement	Joining AI communities, forums, or local meetups can provide opportunities to learn from others, ask questions, and stay updated on the latest developments in the field.

2. Competencies

S/L	Competency	KSA Description
1	**Basic AI Concepts, Programming Curiosity**	**Knowledge:** Understanding fundamental concepts such as machine learning, neural networks, deep learning, natural language processing, and computer vision. **Skills:** Proficiency in a programming language commonly used in AI development, such as Python, along with basic programming concepts and syntax. **Attitudes:** A curious mindset to explore and learn about new AI concepts, techniques, and applications.
2	**AI Applications, Data Handling**	**Knowledge:** Knowledge of various real-world applications of AI across industries such as healthcare, finance, marketing, autonomous vehicles, etc. **Skills:** Ability to manipulate and analyze data using libraries like pandas, NumPy, and scikit-learn, including tasks like data cleaning, feature extraction, and visualization. **Attitudes:** Willingness to learn from failures and mistakes, and continuously improve skills and knowledge in AI.
3	**Ethical Considerations, Machine Learning Basic**	**Knowledge:** Awareness of ethical issues in AI, including bias, fairness, transparency, privacy, and accountability. **Skills:** Basic knowledge of machine learning algorithms, including supervised learning, unsupervised learning, and evaluation metrics. **Attitudes:** Consciousness of ethical implications in AI development and deployment, and a

		commitment to responsible and ethical AI practices.
4	**AI Tools and Technologies, Problem-Solving**	**Knowledge:** Familiarity with popular AI development tools, libraries, and frameworks like TensorFlow, PyTorch, scikit-learn, etc. **Skills:** Skill in formulating AI problems, selecting appropriate algorithms, and implementing solutions to address specific tasks or challenges. **Attitudes:** Flexibility to adapt to changes and advancements in AI technologies and methodologies.
5	**Data Fundamentals, Critical Thinking**	**Knowledge:** Understanding of data types, structures, and preprocessing techniques relevant to AI, including data cleaning, transformation, and feature engineering. **Skills:** Ability to critically evaluate AI solutions, identify potential biases or limitations, and propose improvements or alternatives. **Attitudes:** Readiness to collaborate with others, share knowledge, and work in interdisciplinary teams to solve AI-related problems.

3. Syllabus

INTRODUCTION TO ARTIFICIAL INTELLIGENCE
SEMESTER – IV

Course Code	**M23BCS405**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(2:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Gain a historical perspective of AI and its foundations.
- Become familiar with basic principles of AI toward problem solving
- Get to know approaches of inference, perception, knowledge representation, and learning

Module -1

Introduction: Artificial Intelligence, The Foundations of Artificial Intelligence, History of Artificial Intelligence, The State of the Art in AI.

Intelligent Agent: Agents and Environments, Good Behaviour: Concept of Rationality, The Nature of Environments, The Structure of Agents.

Problem Solving: Problem-Solving Agents, Example Problems.

Chapter 1, Chapter 2, Chapter 3

Module -2

Problem Solving: Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.

Logical Agents: Knowledge–Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving, Effective Propositional Model Checking, Agent Based on Propositional Logic. Chapter 3, Chapter 7

Module -3

First Order Logic: Representation Revisited, Syntax and Semantics of First Order Logic, Using First Order Logic, Knowledge Engineering in First Order Logic.

Inference in First Oder: Propositional vs. First Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Chapter 8, Chapter 9

Module -4

Uncertain Knowledge and Reasoning: **Quantifying Uncertainty**: Acting under Uncertainty, Basic Probability Notation, Inference using Full Joint Distributions, Independence, Baye's Rule and Its Use, Wumpus World Revisited.

Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, Regression and

Classification with Linear Models, Artificial Neural Networks, Support Vector Machines. Chapter 13, Chapter 18
Module -5
Natural Language Processing: Language Model, Text classification, Information Retrieval and Extraction. Case Study: NLP Techniques **Perception**: Image Formation, Early Image Processing Operation, Object Recognition by Appearance, Reconstructing the 3D World.Case Study: Image Processing In Agriculture **Robotics:** Introduction, Robot Hardware, Robotic Perception, Robotic Software Architecture, Application Domain**.** Case Study: Robotic Cars Chapter 22, Chapter 24, Chapter 25
Text Books:
1.Stuart J. Russell and Peter Norvig, Artificial Intelligence, 3rd Edition, Pearson,2015 2.Elaine Rich, Kevin Knight, Artificial Intelligence, 3rd edition, Tata McGraw Hill,2013
Reference Books:
1. R. B Mishra, Artificial intelligence PHI Learning Pvt. Ltd., 2010 2. M. Tim Jones, “Artificial Intelligence: A Systems Approach (Computer Science)”,Jones and Bartlett Publishers, Inc.; 1st Edition, 2008.
Web links and Video Lectures (e-Resources) **1. https://nptel.ac.in/courses/106/105/106105077/** **2. https://archive.nptel.ac.in/courses/106/105/106105152/** **3. https://archive.nptel.ac.in/courses/106/105/106105158/** **4. https://archive.nptel.ac.in/courses/117/105/117105135/** **5. https://archive.nptel.ac.in/courses/107/106/107106090/**

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Introduction, Intelligent Agents, Problem-solving - I	**Competency:** Foundations of Artificial Intelligence **Knowledge:** Provides a foundational understanding of AI concepts and its evolution **Skills:** Algorithm design and programming, essential for developing intelligent systems
2	Week 3-4: Problem Solving - II	**Competency:** Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) search strategies, Heuristic functions **Knowledge:** Provides knowledge about the principles and algorithms underlying these strategies, enabling effective application in AI systems. **Skills:** problem-solving through searching for solutions using both uninformed and informed search strategies, and developing heuristic functions to guide efficient search processes.
3	Week 5-6: Logical Agent, First Order Logic	**Competency:** Knowledge–based agents, : Representation Revisited, Syntax and Semantics of first order logic **Knowledge:** Formal logic principles, predicate calculus, and methods for representing and manipulating knowledge, enabling the design of more robust and interpretable systems. **Skills:** creating intelligent systems capable of logical reasoning and decision-making
4	Week 7-8: Inference in First	**Competency:** Propositional vs. First order inference, Unification and Lifting, Basic

	Order Logic, Quantifying Uncertainty	Probability Notation, Inference using Full Joint Distributions **Knowledge:** Drawing logical conclusions from a set of premises, including resolution, unification, and theorem proving techniques **Skills:** Reasoning and problem-solving, capable of intelligent decision-making and knowledge representation.
5	Week 9-10: Application of AI: Natural Language Processing, Computer Vision, Robotics.	**Competency:** **Building AI application on** **Knowledge:** data preprocessing, feature engineering, model selection, evaluation metrics, and deployment strategies **Skills:** identifying suitable AI techniques and algorithms for solving real-world problems across various domains such as healthcare, finance, marketing, and robotics

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Structured Curriculum	Develop a well-structured curriculum that covers fundamental AI concepts, including machine learning, neural networks, natural language processing, and computer vision. Outline learning objectives, topics, and milestones to guide students through the course.
2	Active Learning Techniques	Incorporate active learning techniques such as problem-based learning, case studies, and group discussions to engage students and encourage participation. Provide opportunities for students to apply AI concepts to real-world problems through projects and assignments.
3	Hands-on Projects	Offer hands-on projects where students can experiment with AI algorithms, tools, and datasets. Provide access to relevant software and resources, such as Python programming environments, AI libraries, and cloud computing platforms.
4	Guest Lectures and Industry Connections	Invite guest speakers from industry, academia, and research institutions to share their expertise and experiences in AI. Organize field trips, industry visits, or virtual seminars to expose students to real-world AI applications and career opportunities.
5	Interactive Lectures and Demonstrations	Use a variety of teaching methods, including interactive lectures, demonstrations, and multimedia presentations, to explain complex AI concepts. Use visual aids, simulations, and interactive tutorials to illustrate key concepts and algorithms.
6	Student-Centered Learning	Empower students to take ownership of their learning by encouraging independent inquiry, research, and exploration. Provide opportunities for self-directed learning through online resources, tutorials, and project-based learning platforms.
7	Assessment and Feedback	Implement a variety of assessment methods, including quizzes, exams, projects, and presentations, to evaluate students' understanding of AI concepts and their ability to apply them. Provide constructive feedback to help students improve their skills and knowledge.
8	Ethical and Social Implications	Integrate discussions on the ethical and social implications of AI into the curriculum. Encourage students to critically evaluate the impact of AI on society, privacy, bias, fairness, and employment.
9	Peer Learning and Collaboration	Foster a collaborative learning environment where students can work together in teams, share ideas, and collaborate on projects. Encourage peer-to-peer learning, mentorship, and peer review to promote knowledge sharing and teamwork.

10	Continuous Improvement and Updates	Continuously update the course content and teaching materials to reflect the latest advancements in AI research, technologies, and applications. Seek feedback from students and colleagues to identify areas for improvement and innovation.

6. Continuous Internal Evaluation:

The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Professional Course (PC)

	Components	Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester-End Examination

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understand the foundational concepts	Understand the foundational concepts and historical development of artificial intelligence, including the principles of intelligent agents, problem-solving strategies, and the evolution of AI technologies.
2	Develop proficiency in knowledge representation and reasoning techniques	Develop proficiency in knowledge representation and reasoning techniques, including predicate logic, ontologies, and common-sense reasoning, to effectively model and solve complex problems in AI applications.
3	Gain practical knowledge and skills in machine learning fundamentals	Gain practical knowledge and skills in machine learning fundamentals, including supervised and unsupervised learning algorithms, evaluation metrics, and techniques for mitigating issues such as overfitting and regularization.
4	Explore the principles and architectures of neural networks and deep learning models	Explore the principles and architectures of neural networks and deep learning models, including perceptron's, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), and understand their applications in various domains such as computer vision and natural language processing
5	Apply Python programming skills to implement AI algorithms and frameworks	Apply Python programming skills to implement AI algorithms and frameworks introduced throughout the course, enabling students to develop hands-on experience in building AI systems and applications
6	Analyze and discuss real-world applications of artificial intelligence across diverse domains	Analyze and discuss real-world applications of artificial intelligence across diverse domains, including robotics, healthcare, and ethical considerations, to understand the societal impact and ethical implications of AI technologies.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description

M23BCS405.1	**Understand and Apply knowledge of AI fundamentals and Intelligent agent types.**
M23BCS405.2	**Apply the use of logic and knowledge representation for problem solving.**
M23BCS405.3	**Formulate knowledge reasoning using propositional logic and first order logic**
M23BCS405.4	**Analyze Quantifying uncertainty using probability notations.**
M23BCS405.5	**Design and implement neural networks for AI applications.**

CO-PO-PSO Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
M23BCS405.1		**3**											**3**	**3**
M23BCS405.2		**3**											**3**	**3**
M23BCS405.3			**3**										**3**	**3**
M23BCS405.4			**3**										**3**	**3**
M23BCS405.5					**3**								**3**	**3**
M23BCS405		**3**	**3**		**3**								**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					
Module 2		**10**				
Module 3			**10**			
Module 4				**10**		
Module 5					**10**	
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					
Module 2		**20**				
Module 3			**20**			
Module 4				**20**		
Module 5					**20**	
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

- Innovation and Advancements: As students become introduced to AI concepts at an early stage, they'll contribute to a culture of innovation and drive advancements in AI technology. This can lead to breakthroughs in areas such as healthcare, transportation, finance, and more, solving complex problems and improving quality of life.
- Workforce Readiness: Introducing AI in education ensures that students are equipped with the knowledge and skills needed to thrive in a future where AI is ubiquitous. This prepares them for AI-related jobs across various sectors, ranging from data science and machine learning engineering to AI ethics and policy-making.
- Ethical AI Development: Education on AI ethics and responsible AI practices cultivates a generation of professionals who prioritize ethical considerations in AI development and deployment. This includes addressing biases, ensuring transparency and accountability, and promoting fairness and inclusivity in AI systems.

- Cross-disciplinary Collaboration: Introduction to AI fosters collaboration across different disciplines, as AI intersects with fields such as computer science, mathematics, engineering, psychology, sociology, and more. Collaborative efforts lead to innovative solutions that tackle complex challenges from multiple perspectives.
- Entrepreneurship and Startups: Students introduced to AI may be inspired to pursue entrepreneurship and create AI-driven startups, addressing niche markets or disrupting existing industries. This entrepreneurial spirit contributes to economic growth, job creation, and technological innovation.
- AI Education Accessibility: Advancements in AI education technologies, such as online courses, interactive tutorials, and AI-driven personalized learning platforms, make AI education more accessible to learners worldwide. This democratization of AI education empowers individuals from diverse backgrounds to acquire AI skills and knowledge.
- Global Impact: Introduction to AI transcends geographical boundaries, empowering learners from different regions to contribute to global AI initiatives. Collaboration among international institutions, researchers, and students accelerates AI research, innovation, and knowledge-sharing on a global scale.
- AI for Social Good: Educating students on the potential of AI for social good encourages them to apply AI technologies to address pressing societal challenges, such as healthcare disparities, environmental sustainability, education accessibility, and poverty alleviation. AI-driven solutions have the potential to create positive social impact and promote inclusive development.
- Continuous Learning and Adaptation: In a rapidly evolving field like AI, continuous learning and adaptation are essential. Introduction to AI instills a culture of lifelong learning, encouraging individuals to stay updated with the latest advancements, trends, and best practices in AI throughout their careers.
- Ethical Leadership and Governance: As future leaders and policymakers, students introduced to AI play a crucial role in shaping ethical AI governance frameworks and policies. They advocate for regulations that ensure AI technologies are developed and deployed responsibly, balancing innovation with ethical considerations and societal well-being.

4th Semester	**Professional Course Lab (PCL)** **ANALYSIS & DESIGN OF ALGORITHMS LABORATORY**	**M23BCSL406**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Knowledge of Algorithms**	Students should have a good grasp of fundamental algorithms and data structures. This includes understanding sorting algorithms (like bubble sort, merge sort, quicksort), searching algorithms (linear search, binary search), and data structures (arrays, linked lists, trees, graphs).
2	**Programming Proficiency**	Proficiency in a programming language is essential. Most often, labs might require programming in languages like C, C++, Java, Python, or similar. Students should be comfortable with implementing algorithms and data structures in at least one of these languages.
3	**Mathematical Foundations**	A strong foundation in discrete mathematics is beneficial. Topics such as combinatorics, probability, graph theory, and basic number theory are often used in the analysis of algorithms. Understanding mathematical concepts allows students to analyze algorithm efficiency and correctness rigorously.
4	**Algorithm Analysis**	Understanding how to analyze the time complexity (Big-O notation) and space complexity of algorithms is crucial. This involves being able to predict and evaluate the efficiency of algorithms based on their input size.
5	**Basic Data Structures**	Familiarity with common data structures (arrays, linked lists, stacks, queues, trees, graphs) and their operations (insertion, deletion, traversal) is necessary. Labs might involve implementing algorithms that utilize these data structures effectively.

2. Competencies

S/L	Competency	Description
1	**Foundational Algorithmic Analysis and Design**	**Knowledge of Algorithms**: Understanding the fundamentals of the analysis of algorithm efficiency and learn to conduct mathematical analysis for both non-recursive and recursive algorithms. Understanding basic algorithmic strategies and their applications to simple problem-solving scenarios. **Skills:** Able to evaluate algorithm performance using analysis frameworks, asymptotic notations, and efficiency classes. **Attitudes:** Critical and analytical attitude and adopt a thorough and detail-oriented mindset to analyze more complex scenarios.
2	**Algorithmic Strategy and Problem-Solving**	**Knowledge:** Gain insights into solving complex problems through comprehensive, albeit computationally expensive, methods. Understanding how to reduce problem size to achieve solutions efficiently. Imparting knowledge on breaking down problems into smaller, more manageable sub problems, solving them independently, and combining their solutions. **Skills:** Develop skills in systematically exploring all possible solutions to find the optimal one. Ability to simplify complex problems by reducing their size. Proficiency in breaking down problems into smaller sub problems, solving them independently, and efficiently combining the results. **Attitudes:** Analytical thinking and adaptability, step-by-step problem-solving mentality
3	**Algorithm Optimization**	**Knowledge:** Understanding of advanced optimization algorithms. Learn to balance and optimize the use of computational resources. **Skills:** Efficient organization and manipulation of optimize algorithm performance. Ability to analyze and optimize algorithms by making strategic tradeoffs between memory usages and processing speed.

		Attitudes: Analytical thinking and adaptability, step-by-step problem-solving mentality
4	**Algorithmic Problem-Solving and Optimization**	**Knowledge:** Understanding the construction of optimal solutions by making locally optimal choices at each step, enabling efficient problem-solving in various domains. **Skills:** Proficiency in applying algorithms real time problems. Ability to make locally optimal choices at each step to construct globally optimal solutions, enhancing their problem-solving skills across various domains, including graph theory and data compression. **Attitudes: Problem-solving skills**and **ability to optimize solutions** by breaking down complex problems into smaller sub problems. **Strategic decision-making**and**efficiency.**
5	**Advanced Problem-Solving and Algorithmic Analysis**	**Knowledge:** Understanding the theoretical boundaries of algorithmic efficiency and the inherent difficulty of certain types of computational problems**.** **Skills:** Critical thinking skills to understand the theoretical boundaries of computational problem-solving, as well as the ability to analyze the complexity of problems. **Attitudes:** Perseverance and adaptability, to tackle challenging problems through iterative exploration, strategic optimization.

3. Syllabus

Analysis & Design of Algorithms Laboratory
SEMESTER – IV

Course Code	**M23BCSL406**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course objectives: This course will enable students to:

- To design and implement various algorithms in C/C++ programming using suitable development tools to address different computational challenges.
- To apply diverse design strategies for effective problem-solving.
- To Measure and compare the performance of different algorithms to determine their efficiency and suitability for specific tasks.

PRACTICAL COMPONENT
Implement all the programs in C Programming Language

1.	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirectedgraphusingKruskal'salgorithm.
2.	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirectedgraphusingPrim'salgorithm.
3.	Design and implement C/C++ProgramtosolveAll-PairsShortestPathsproblemusingFloyd'salgorithm. DesignandimplementC/C++ProgramtofindthetransitiveclosureusingWarshal'salgorithm.
4.	DesignandimplementC/C++Programtofindshortestpathsfromagivenvertexinaweighted connectedgraphtootherverticesusingDijkstra'salgorithm.
5.	DesignandimplementC/C++ProgramtoobtaintheTopologicalorderingofverticesinagiven digraph.
6.	DesignandimplementC/C++Programtosolve0/1KnapsackproblemusingDynamic Programmingmethod.
7.	DesignandimplementC/C++ProgramtosolvediscreteKnapsackandcontinuousKnapsack problemsusinggreedyapproximationmethod.
8.	DesignandimplementC/C++ProgramtofindasubsetofagivensetS={sl,s2,. .... ,sn}o fnpositive integerswhosesumisequaltoagivenpositiveintegerd.
9.	Design and implement C/C++ Program to sort a given set of n integer elements using SelectionSort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
10.	Design and implement C/C++ Program to sort a given set of n integer elements using Quick

	Sortmethod and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versusn. The elements can be read from a file or can be generated using the random number generator.
11.	Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sortmethod and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
12.	DesignandimplementC/C++ProgramforNQueen'sproblemusingBacktracking.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	Program1, Program2.
2	Week 3-4:	Program3, Program4.
3	Week 5-6:	Program5, Program6.
4	Week 7-8:	Program7, Program8.
5	Week 9-10:	Program9, Program10.
6	Week 11-12:	Program11, Program12.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Lectures	Use interactive lectures to introduce new concepts. Incorporate questions and discussions to engage students.
2	Coding Sessions:	Demonstrate the implementation of different algorithms live, showing step-by-step coding and debugging.
3	Lab Exercises	Design lab exercises that require students to implement and manipulate data structures.
4	Coding Assignments	Assign regular coding tasks that reinforce lecture material and provide practical experience.
5	Group Projects	Encourage students to work in groups for larger projects, fostering teamwork and collaborative problem-solving.
6	Code Documentation	Practice writing clear and comprehensive documentation for all coding assignments and projects.

6. Assessment Details (both CIE and SEE)

Marks distribution for Program based Practical Course for CIE

Sl. No.	Description	% of Marks	In Marks
1	Observation, write-up, algorithm/program/execution	80% of the maximum	80
2	Viva-Voce	20% of the maximum	20
Total		100%	100

Marks scored by the student for 100 are scaled down to 50 marks.

SEE for practical Course (Irrespective of Experiment or program based):

1. SEE marks for practical course shall be 50 marks

2. Marks distribution for Experiment based Practical Course for Final CIE

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. See for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Algorithm Fundamentals**	Students will comprehend the definition and significance of algorithms in problem-solving.
2	**Analyzing Algorithm Efficiency**	Students will apply a structured framework to assess algorithmic performance in terms of time and space complexity.
3	**Mastering Brute Force Approaches**	Students will gain proficiency in implementing and evaluating brute force algorithms like Selection Sort, Bubble Sort, and Sequential Search.
4	**Exploring Various Algorithmic Strategies**	Students will explore and apply strategies like Merge Sort, Insertion Sort, Balanced Search Trees, and Heapsort.
5	**Advanced Algorithmic Techniques**	Students will gain expertise in implementing Dynamic Programming algorithms for problems like the Knapsack Problem, and Greedy Method for tasks like Prim's Algorithm and Huffman Trees
6	**Understanding Algorithmic Limitations and Coping Strategies**	Students will identify limitations of algorithms including NP-Completeness, and apply coping strategies like Backtracking, Branch-and-Bound, and Approximation Algorithms
7	**Synthesizing Knowledge and Applying in Problem Domains**	Students will synthesize their understanding of various algorithmic techniques to tackle complex problems in real-world scenarios, demonstrating proficiency in algorithm design and analysis.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

Cos	Description
M23BCSL406.1	Develop programs to solve computational problems using suitable algorithm design strategy.
M23BCSL406.2	Compare algorithm design strategies by developing equivalent programs and observing running times for analysis (Empirical).
M23BCSL406.3	Make use of suitable integrated development tools to develop programs
M23BCSL406.4	Choose appropriate algorithm design techniques to develop solution to the computational and complex problems.
M23BCSL406.5	Demonstrate and present the development of program, its execution and running time(s) and record the results/inferences

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCSL406.1	3												3	3
M23BCSL406.2	3												3	3
M23BCSL406.3		3											3	3
M23BCSL406.4			3										3	3
M23BCSL406.5			3	3	3								3	3
M23BCSL406	3	3	3	3	3								3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**15**	**5**	**10**	**5**	**15**	**50**
Total	**15**	**5**	**10**	**5**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**30**	**10**	**20**	**10**	**30**	**100**
Total	**30**	**10**	**20**	**10**	**30**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The future of data structures lab focuses on equipping students with:

Comprehensive Analysis: Develop a thorough understanding of the characteristics, advantages, and disadvantages of various linear (arrays, linked lists, stacks, queues) and non-linear (trees, graphs) data structures.

Practical Demonstration: Illustrate and implement the functioning of different data structures, showcasing their operations and practical applications in real-world scenarios.

Algorithm Application: Select and employ the most suitable searching (linear search, binary search) and sorting algorithms (bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort) based on specific problem requirements.

Real-World Problem Solving: Apply the appropriate data structure to design and implement solutions for real-world problems, ensuring efficiency, scalability, and maintainability.

4th Semester	**Engineering Science Course (ES)** **GRAPH THEORY**	**M23BCS407A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Graph Theory	Understandingofgraphs, types of graphs and graph Isomorphism Vertex and degree of graph
2.	Linear algebra	Familiarity with linear algebra and basic counting methods such as binomial coefficient is assumed
3.	Combinatorics & Discrete Mathematics	Basic knowledge of combinatorics, probability theory and types of functions
4.	BasicUnderstandingTress and Algorithm	Basicunderstandingof writing Algorithms
5.	FundamentalMathematics Knowledge	Knowledge of basic algebraic mathematics like union intersections permutations and combinations and binomial Theorem.
6.	BasicUnderstandingGenerating Function	Knowledge of generating functions and calculationtechniques, Partitions of Integers
7.	FundamentalMathematics Knowledge	Knowledge of basic Mathematics in solving First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients
8.	Graph Theory	Understandingofgraphs, types of graphs and graph Isomorphism Vertex and degree of graph
9.	Linear algebra	Familiarity with linear algebra and basic counting methods such as binomial coefficient is assumed
10.	Previous Coursework	Completion of introductory courses in Basic electronics or a related field

2. Competencies

S/L	Competency	KSA Description
1	**Graph Theory**	**Knowledge:** Graphs, Euler Trails and Circuits and Hamilton paths Knowledge of Graph Colouring and directed graph **Skills:** Ability to apply graph theory can describe the structure of the circuit by using a directed graph **Attitudes:** Appreciation for the importance of graph theory in Modeling transportation Network Analysis,
2	**Trees & Optimization and Matching**	**Knowledge:** UnderstandingShortestPathAlgorithm, Prefix code **Skills:** Designing Dijkstra‟sShortestPathAlgorithm,MinimalSpanningThe algorithms of Kruskal and Prim, circuits. **Attitudes:**AppreciationfortheroleofShortestPathAlgorithminTransport Networks
3	**Fundamental Principles of Counting**	**Knowledge:** Understanding of Rules of Sum and Product **Skills:** Permutations and Combinations, Binomial Theorem **Attitudes:** Valuing the importance of Permutations and Combinations to create strong passwords and secure system
4	**Generating Functions:**	**Knowledge:** Understanding the Calculational Techniques, Partitions of Integers and summation operator **Skills:** ApplyingExponential GeneratingFunction to solve the linear recurrence relation **Attitudes:** Solve the Linear recurrence problem
5	**Recurrence Relations**	**Knowledge:** UnderstandingFirst and Second Order Linear Recurrence Relation **Skills:** Recurrence relations are used to reduce complicated problems to an iterative process based on simpler version of the problem **Attitudes:** Learning of analysis of Algorithm

3. Syllabus

Graph Theory
SEMESTER – IV

Course Code	**M23BCS407A**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

Appreciate the definition and basics of graphs along with types and their examples. Understand the notion of planarity and coloring of a graph

Understand the definition of a tree and learn its applications to fundamental circuits. Know the applications of graph theory to network flows

To give the learner a broad exposure of combinatorial Mathematics through applications especially the Computer Science application

Module -1

Introduction to Graph Theory: Definition and properties of a graph, sub graph, and Examples, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles, Graph Colouring, and Chromatic Polynomials. Directed graphs and their properties.

Module -2

Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes.**Optimization and Matching:** Dijkstra‟s Shortest Path Algorithm, Minimal Spanning Trees – The algorithms of Kruskal's and Prim, Transport Networks – Max-flow, Min-cut Theorem. Matching Theory.

Module -3

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition.The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle,

Module -4

Generating Functions: Introductory Examples, Definition and Examples – Calculational Techniques, Partitions of Integers, the Exponential Generating Function. The Summation Operator.

Module -5

Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients, The Non-homogeneous Recurrence Relation. The Method of Generating Functions.

Text Books:

1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education, 2004.
2. Narasingh Deo: Graph Theory with Applications to Engineering and Computer Science. 2016.

ReferenceBooks

1. Kenneth H Rosen: Discrete Mathematics & its Applications with Combinatorics and Graph Theory. 6th Edition, 2009.
2. D.S. Chandrasekharaiah: Graph Theory and Combinatorics, Prism, 2005.
3. Chartrand Zhang: Introduction to Graph Theory, TMH, 2006.
4. Richard A. Brualdi: Introductory Combinatorics, 4th Edition, Pearson Education, 2004.
5. Geir Agnarsson & Raymond Geenlaw: Graph Theory, Pearson Education, 2007.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Definition and properties of a Graph	Definition and properties of a graph, sub graph, and Examples, Complements Graph Isomorphism, Vertex Degree, Euler Trails and Circuits Planar Graphs, Hamilton Paths and Cycles Graph Coloring and Chromatic Polynomials Directed graphs and their properties.
2	Week 3-4: Trees and	Definitions, and Properties of tress, Routed Trees, Trees and Sorting

	Optimization and Matching:	Weighted Trees and Prefix Codes. Optimization and Matching- Dijkstra‟s Shortest Path Algorithm, Minimal Spanning Trees – The algorithms of Kruskal and Prim, Transport Networks Max-flow, Min-cut Theorem. Matching Theory.
3	Week 5-6: Fundamental Principles of Counting & The Principle of Inclusion and Exclusion	The Rules of Sum and Product Permutations, Combinations The Binomial Theorem, Combinations with Repetition. The Principle of Inclusion and Exclusion Generalizations of the Principle,
4	Week 7-8: Generating Functions:	Generating Functions- Introductory Examples Definition and Examples – Calculation Techniques Partitions of Integers Exponential Generating Function The Summation Operator.
5	Week 9-10: Recurrence Relations	First Order Linear Recurrence Relation The Second Order Linear Recurrence Relation Homogeneous Recurrence Relation with Constant Coefficients The Non-homogeneous Recurrence Relation The Method of Generating Functions.
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice

1. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of graph theory concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Engineering Science Course (ES)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10

	TotalMarks	**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester-End Examination

5. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
6. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
7. The students have to answer 5 full questions selecting one full question from each module.
8. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	UnderstandingDefinitions and types of Graphs	Students will learn Use Graph theory in modeling transportation networks, including road networks, railway systems, and flight routes. Traffic optimization and resource allocation by analyzing the connectivity and distances between locations within the network.
2	Designing Huffman Coding	Studentswilllearntodesign the Huffman code with the help of trees, routed tress and Prefix codes
3	Proficiency inPrefix code	Students will become proficient in writing Prefix code, Dijkstra's Shortest path algorithm and the algorithms of kruskal and prism
4	Project-Based Learning	Through hands-on projects, students will apply their knowledge ofMake use Dijkstra's Shortest path algorithm, transport networks.
5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS407A.1	Apply the concepts of graph theory, trees , generating functions and recurrence relation to solve engineering problems.
M23BCS407A.2	Analyze Computer Science Engineering application problems through Dijkstra's Shortest path algorithm, recurrence relation .
M23BCS407A.3	Demonstrate an understanding of the fundamental concepts of graph theory, digraphs, trees, finding Paths and cycles, weighted graphs matching andgraph coloring.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS407A.1	**3**												**3**	
M23BCS407A.2		**3**											**3**	
M23BCS407A.3	**3**													**3**
M23BCS407A	**3**	**3**											**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	**Total**
Module 1	**2**	**5**	**3**	**10**
Module 2	**2**	**5**	**3**	**10**
Module 3	**2**	**5**	**3**	**10**
Module 4	**2**	**5**	**3**	**10**
Module 5	**2**	**5**	**3**	**10**
Total	10	**25**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	**Total**
Module 1	**4**	**10**	**6**	**20**
Module 2	**4**	**10**	**6**	**20**
Module 3	**4**	**10**	**6**	**20**
Module 4	**4**	**10**	**6**	**20**
Module 5	**4**	**10**	**6**	**20**
Total	**20**	**50**	**30**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Graph Theory" course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of networks. Here are some notable contributions

Cryptography: The course contributes to the understanding of algorithms using paths in any graph and block encryption algorithms using directed graphs an encryption method in which a graph is the key. Encryption is done by charting a path on that graph. A sequence of vertices in the path of the key graph forms the plain text. A sequence of edges between those vertices forms the cipher text. The girth of a simple graph G is the length of its shortest cycle. Simple graphs of large girth turn out to be useful in networking, error correction theory, Cryptography and other problems of Computer Science.

Computer Engineering: Shortest path algorithms have many applications. As noted earlier, mapping software like Google or Apple maps makes use of shortest path algorithms. They are also important for road network, operations, and logistics research. Shortest path algorithms are also very important for computer networks, like the Internet.

Discrete Mathematics: The discrete nature of sequences prevents us from using calculus on sequences. A generating function is a continuous function associated with a given sequence. For this reason, generating functions are very useful in analyzing discrete problems involving sequences of numbers or sequences of functions. Find a closed formula for a sequence given in a recurrence relation. For example, consider Fibonacci numbers. Find recurrence relations for sequences—the form of a generating function

Industry Applications: Permutations and Combinations help the management of a company make decisions by minimizing the variables in selecting the profitable option from the available objects to proceed with business activities. The study of permutation and combination is essential in developing games or scenario-based activities as it allows game developers to understand the possibilities of certain events occurring within the game or activity

4th Semester	Engineering Science Course (ES) LINEAR ALGEBRA	M23BCS407B

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Vector Algebra	Understanding of Vector Algebra.
2.	Matrix Theory	Knowledge of matrix theory to perform row operations on matrices and find bases and dimension of vector spaces and analyze whether a system is consistent or inconsistent and its solution is unique or infinite.
3.	Calculus	Proficiency in Calculus to mapping known as a linear map is one that maintains the addition and multiplication of vectors throughout the process.
4.	Geometry	Basic understanding of programming concepts, as Verilog is a hardware description language (HDL) used for digitaldesign.
5.	Group Theory	Familiar with real and complex numbers, and understand the concept of functions.
6.	Fundamental Mathematics Knowledge	Knowledge of addition, multiplications, matrix multiplications Knowledgeof Representasystemintheformoflinearequations.
7.	Previous Coursework	CompletionofintroductorycoursesinLinear Algebraorarelatedfield

2. Competencies

S/L	Competency	KSA Description
1	**Linear Equations**	**Knowledge:** Understanding Linear equations, Matrix Operations, LU-decomposition **Skills**: Ability to apply matrix operations to solve the system of equations **Attitudes:** Appreciation for represent a system in the form of linear equations
2	**VectorSpaces**	**Knowledge:** Vector spaces, subspaces, Bases and Dimension **Skills:** Identify vector spaces and subspaces **Attitudes:** Appreciationforperform row operations on matrices and find bases and dimension of vector spaces.
3	**Linear Transformations**	**Knowledge:** Algebra of Linear Transformation, Inverse of a linear transformation **Skills:** Apply Linearly transform the system from one dimension to another and represent the pertinent linear transformation in matrix form. **Attitudes:**Linearly transform the system from one dimension to another and represent the pertinent linear transformation in matrix form.
4	**InnerProductSpaces**	**Knowledge:** Orthogonal sets, Gram-Schmidt Process **Skills:** Analyze image and signal processing problems **Attitudes:** AppreciationforAnalyze image and signal processing problems
5	**SymmetricMatricesand QuadraticForms**	**Knowledge**: Matrix Operations, Diagonalizations, Quadratic forms **Skills:** Singular value decomposition for problems arising in power/control system analysis, signals and systems **Attitudes:** Solving singular value decomposition for problems arising in power/control system analysis, signals and systems

3. Syllabus

Linear Algebra
SEMESTER – IV

Course Code	**M23BCS407B**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

1. Represent a system in the form of linear equations.
2. Find the solution of the system of linear equations using matrix operations.
3. Identify vector spaces and subspaces.
4. Transform a vector space of one dimension into another.
5. Factorize a given matrix using different methods.

Module -1

Fields; system of linear equations, and its solution sets; elementary row operations and echelon forms; matrix operations; invertible matrices, LU-decomposition.

Module -2

Vector Spaces: Vector spaces; subspaces; bases and dimension; coordinates; summary of row-equivalence; computations concerning subspaces.

Module -3

Linear Transformations: Linear transformations; algebra of linear transformations; isomorphism; representation of transformations by matrices; linear functional; inverse of a linear transformation.

Module -4

Inner Product Spaces: Inner products; inner product spaces; orthogonal sets and projections; Gram-Schmidt process; QR-factorization.

Module -5

Symmetric Matrices and Quadratic Forms: Diagonalization; quadratic forms; constrained optimization; Singular value decomposition.

Text Books:

1.David C. Lay, “Linear Algebra and its Applications,” 3rd edition, Pearson Education (Asia) Pte. Ltd, 2005.
2. Kenneth Hoffman and Ray Kunze, "Linear Algebra," 2nd edition, Pearson Education (Asia) Pte. Ltd/2004.

ReferenceBooks

1. Bernard Kolman and David R. Hill, "Introductory Linear Algebra with Applications”, Pearson Education (Asia) Pte. Ltd, 7th edition, 2003.
2.Gilbert Strang, "Linear Algebra and its Applications”, 3rd edition, Thomson Learning Asia, 2003

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Linear Equations	Introduction to Fields System of linear equations and its solution Sets- elementary row operations and echelon forms Matrix operations Invertible matrices LU-decomposition
2	Week 3-4: Vector Spaces	Introduction to Vector spaces and subspaces Bases and dimension Coordinates; summary of row-equivalence Computations concerning subspaces. Worked problems
3	Week 5-6: Linear Transformations:	Linear transformations; algebra of linear transformations Isomorphism; representation of transformations by matrices Linear functional Worked problems Inverse of a linear transformation
4	Week 7-8: Inner Product	Introduction to Inner Product Spaces

	Spaces:	Orthogonal sets and projections Gram-Schmidt process and Worked problems QR-factorization. Worked problems
5	Week 9-10: Symmetric Matrices and Quadratic Forms	Symmetric Matrices and Quadratic Forms Diagonalization; quadratic forms Worked problems Constrained optimization Singular value decomposition
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice with assignments

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of the concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Engineering Science Course (ES)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester-End Examination

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding	Students will learn competence with the basic ideas of linear algebra including

	Linear Equations, Matrices and Vector Space	concepts of linear systems, independence, theory of matrices, linear transformations, bases and dimension, eigenvalues, eigenvectors and diagonalization.
2	Designing and Analyze the Mathematical Statement	StudentswilllearntoAnalyzewhetherasystemisconsistentorinconsistentandits solutionisuniqueorinfinite. Designing logical progressions of precise mathematical statements to justify and communicate your reasoning
3	Proficiency in Matrix	Students will become proficient in writing a matrix and Perform row operations on matrices and find bases and dimension of vector space.
4	Project-BasedLearning	Through hands-on projects, students will apply their knowledge in techniques of constrained optimization and singular value decomposition for problems arising in power/control system analysis, signals and systems.
5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

2. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS407B.1	Apply the concepts of vector space, linear transformation and inner product space to solve the engineering problem
M23BCS407B.2	Demonstrate the importance of vector space inner, product space and linear transformations in computer science engineering
M23BCS407B.3	Analyze the computer science engineering application problems through linear transformation

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS407B.1	**3**												**3**	
M23BCS407B.2		**3**												**3**
M23BCS407B.3	**3**												**3**	
M23BCS407B	**3**	**3**											**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**2**	**5**	**3**			**10**
Module 2	**2**	**5**	**3**			**10**
Module 3	**2**	**5**	**3**			**10**
Module 4	**2**	**5**	**3**			**10**
Module 5	**2**	**5**	**3**			**10**
Total	10	**25**	**15**			**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**4**	**10**	**6**			**20**

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

Module 2	**4**	**10**	**6**			**20**
Module 3	**4**	**10**	**6**			**20**
Module 4	**4**	**10**	**6**			**20**
Module 5	**4**	**10**	**6**			**20**
Total	**20**	**50**	**30**			**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10.Future with this Subject

The "Linear Algebra" course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

Data classification in Internet of Things: The amount of information increases explosively in Internet of Things, because more and more data are sensed by large amount of sensors. The explosive growth of information makes it difficult to access information efficiently, so it is an effective method to decrease the amount of information to be transferred on network by text classification. The text classification algorithm based on vector space model. This algorithm improves the feature selection and weighting methods by introducing synonym replacement to traditional text classification algorithms. The experimental results show that the proposed classification algorithm has considerably improved the precision and recall of classification

Image processing: Matrices are extensively used in image processing for operations such as filtering, smoothing, scaling, and rotating images. In image processing, an image can be represented as a matrix of pixels, where each pixel corresponds to an element in the matrix.

Machine learning and artificial intelligence: Matrices are used in machine learning and artificial intelligence for data representation, modeling, and training of algorithms such as neural networks, support vector machines, and principal component analysis

Data Science: The QR factorization is one of these matrix factorizations that is very useful and has very important applications in Data Science, Statistics, and Data Analysis. One of these applications is the computation of the solution to the Least Squares (LS) Problem. The Gram-Schmidt process is a collection of procedures that converts a collection of linearly independent vectors into a collection of orthonormal vectors that cover the same space as the original set.

Computer Graphics: Linear transformations are widely used in various fields such as computer graphics, where they are essential for rendering 3D objects onto a 2D screen. They are also crucial in solving systems of linear equations, which is a fundamental problem in numerous scientific and engineering disciplines

4th Semester	**Engineering Science Course (ES)** **STATISTICAL AND NUMERICAL METHODS**	**M23BCS407C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Combinatorics & Discrete Mathematics	Basic knowledge of Combinatorics, probability theory and types of functions
2.	Linear algebra	Familiarity with linear algebra and basic counting methods such as binomial coefficient is assumed
3.	Mathematics	ProficiencyinalgebraforBoolean expressionsimplification usingK-map Techniques
4.	FundamentalMathematics Knowledge	Knowledge of basic algebraic mathematics like union intersections permutations and combinations and binomial Theorem.
5.	Relations and Functions	AbilitytoanalyzeCartesian product of set and identify the relations
6.	Algebra	Proficiency in algebraic manipulations, factorization techniques, and solving algebraic equations is necessary for dealing with functions effectively.
7.	Matrices and Determinants:	While not directly related to functions, knowledge of matrices and determinants can be helpful in certain types of function problems.
8.	Probability and Statistics:	Understanding basic probability concepts and statistics can be useful in certain types of function problems that involve probability distributions or data analysis.
9.	Previous Coursework	Completion of introductory courses in Mathematics or a related field

2. Competencies

S/L	Competency	KSA Description
1.	**Random variable and probability distribution**	**Knowledge:** Understandingthealgorithm development, data analysis, machine learning, and simulation modeling. **Skills:** ApplyingProbability to analyze data analysis, statistical inference, and machine learning **Attitudes:** Valuingtheimportance in decision and estimation problems, and constructs computer algorithms for generating observations from the various distributions
2.	**Design of experiments**	**Knowledge:** Design of experiments, Block Design, Latin square design, Graeco Latin Squares **Skills:**Using statistical theory of the design of experiments **Attitudes:** Appreciationforthe Latin and Graeco-Latin squares have an important application to the statistical theory of the design of experiments.
3.	**Estimation**	**Knowledge:** Statistics Inference, Estimation error-bias **Skills:** Using point estimate definition is a calculation where a sample statistic is used to estimate or approximate an unknown population parameter **Attitudes:** Appreciation for analyzing the interval estimation is the use of sample data to estimate an interval of possible values of a parameter of interest.
4.	**Reliability Engineering**	**Knowledge:** Understandingof Concepts of Reliability, Reliability of systems **Skills:** Applying Reliability engineering can be applied to many business functions, from design to maintenance **Attitudes:** Valuing the importance of reliability is a critical factor that focuses on the ability of a system, product, or process to perform its intended functions without malfunctioning or breaking down consistently.
5.		**Knowledge:** Markov chain and related problems. Queuing theory- Poisson queuing system,

	Stochastic Process	**Skills:** ApplyingStochastic Process to analyze Image Processing, Neuroscience, Bio Informatics, Financial Management, Statistics **Attitudes:** Valuingtheimportance of Stochastic Processes in real-time mathematical model of systems which has a continuous random varying nature
6.	**M/M/1 and M/M/s queuing models.**	**Knowledge:** Little law. Discussion of M/M/1 and M/M/s queuing models. **Skills:** Ability to apply Queuing Theory in model **Attitudes:** Valuingtheimportance of M/M/1 Queue: The M/M/1 queue represents a single-server queuing system with Poisson arrivals, exponentially distributed service times, and a first-come-first-served discipline.

3. Syllabus

Statistical and Numerical Methods
SEMESTER – IV

Course Code	**M23BCS407C**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

1. To familiarize the important tools of advanced numerical methods required to analyze the engineering problems.
2. Acquire the knowledge of probability and statistics applied in their core domain
3. To apply the knowledge of statistical techniques, stochastic process and queuing theory to offer solutions the engineering problems
4. Improve their Mathematical Thinking and acquire skills required for sustained lifelong learning.

Module -1

Probability Distributions : Theoretical distributions: Discrete and continuous random variables
Discrete distributions: Geometric distributions, Hyper geometric distribution and Uniform distribution. Continuous distributions: Uniform Distribution, Gamma distributions, t-distribution, F-distribution and chi-square distribution

Module -2

Design of experiments: Analysis of variance, no way classification, completely Randomized design, randomized Block Design, Latin square design, Graeco Latin Squares

Module -3

ESTIMATION Parameter estimation-Point and interval; Estimation error-bias, variance and risk, Method of moments, Estimator design approach- Maximum Likelihood, confidence interval.

Module -4

Reliability Engineering: Concepts of Reliability, Reliability of systems, Availability of Markovian Systems Availability Function

Module -5

Stochastic Process: Classification of stochastic process with examples. Markov chain and related problems. Queuing theory- Poisson queuing system, Little law. Discussion of M/M/1 and M/M/s queuing models

Text Books:

1. K.F. Riley, M.P.Hobson and S.J. Bence, “Mathematical Methods for Physics and Engineering”, Cambridge University Press 3rd Edition, 2017.
2. E. Kreyszig John Wiley & Sons, “Advanced Engineering Mathematics” 10thEd.,(Reprint), 2017.
3. T.Veerarajan, “Probability, Statistics and Random Process”, Tata Mc-Graw Hill Co. 3rd, Edition, 2016.

References Books

1. S. S. Sastry, “Introductory Methods of Numerical Analysis”, Prentice Hall of India 4th, Edition, 2011.
2. M. K. Jain, S. R. K. Iyengar and R. K. Jain, “Numerical Methods for Scientific and Engineering”, Computation New Age Int. Publishers 6th Edition, 2014.
3. G.R. Grimmet and D.R. Stirzaker, “Probability and Random Processes”, Oxford University Press 3rd Edition, 2001.
4. G. Haribaskaran “Probability, Queueing Theory and Reliability Engineering

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Probability Distributions :	Discrete and continuous random variables Geometric distributions Hyper geometric distribution and Uniform distribution Continuous distributions: Uniform Distribution Gamma distributions, t-distribution F-distribution and chi-square distribution Worked Problems
2	Week 3-4: Design of experiments:	Analysis of variance, no way classification completely Randomized design Randomized Block Design Latin square design Graeco Latin Squares Worked Problems
3	Week 5-6: Estimation	Parameter estimation-Point and interval Estimation error-bias, variance and risk Method of moments Estimator design approach Maximum Likelihood, confidence interval Worked Problems
4	Week 7-8: Reliability Engineering	Concepts of Reliability Reliability of systems Worked Problems Availability of Markovian Systems Availability Function Worked Problems
5	Week 9-10: Stochastic Process	Classification of stochastic process with examples Markov chain and related problems. Queuing theory- Poisson queuing system Little law and Problems Discussion of M/M/1 queuing models. Discussion of M/M/s queuing models.
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-on practice

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Engineering Science Course (ES)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester-End Examination

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	UnderstandingProbability Distributions	Students will learn to calculate confidence intervals for parameters and to calculate critical regions for hypothesis tests. For univariate data, it is often useful to determine a reasonable distributional model for the data
2	Design of experiments	Students will learn to design of experiments, Block Design, Latin square design, Graeco Latin Squares
3	Proficiency inReliability engineering	Students will become proficient in applied to many business functions, from design to maintenance.
4	Project-BasedLearning	Through hands-on projects, students will apply their knowledge of Make use Estimations and Stochastic process
5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS407C.1	Apply the concepts of stochastic process Probability Distributions, estimation and design of experiments to solve the engineering problems
M23BCS407C.2	Demonstrate the importance of Probability Distributions, estimation and stochastic process in Computer Science Engineering
M23BCS407C.3	Analyze the Computer Science Engineering applications problems through probability, stochastic process

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS407C.1	**3**												**3**	
M23BCS407C.2		**3**											**3**	
M23BCS407C.3	**3**													**3**
M23BCS407C	**3**	**3**											**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**2**	**5**	**3**			**10**
Module 2	**2**	**5**	**3**			**10**
Module 3	**2**	**5**	**3**			**10**
Module 4	**2**	**5**	**3**			**10**
Module 5	**2**	**5**	**3**			**10**
Total	**10**	**25**	**15**			**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**4**	**10**	**6**			**20**
Module 2	**4**	**10**	**6**			**20**
Module 3	**4**	**10**	**6**			**20**
Module 4	**4**	**10**	**6**			**20**
Module 5	**4**	**10**	**6**			**20**
Total	**20**	**50**	**30**			**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The" Statistical and Numerical Methods” course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

Numerical analysis: The course contributes to the understanding to solves continuous problems using numeric approximation. It involves designing methods that give approximate but accurate numeric solutions, which is useful in cases where the exact solution is impossible or prohibitively expensive to calculate

Telecommunications and Networking: Probability theory is essential in the design and analysis of communication systems, including wireless networks, telecommunications networks, and the internet. It helps in optimizing resource allocation, managing network congestion, and evaluating system performance.

Mathematical Finance: Probability as a subject in and of itself has rarely been truly appreciated by mathematicians in other disciplines. This has gradually changed over the last 50 years, as occasionally brilliant mathematicians’ show how it can be used to solve, or to explain, and/or to give intuitive content to thorny mathematical issues. We provide some examples and then give a wild speculation as to where the field, at least in Mathematical Finance, might go in the future.

4th Semester	**Engineering Science Course (ES)** **VECTOR SPACE AND METRIC SPACE**	**M23BCS407D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Vector algebra	Understandingof Basics of Vector Algebra.
2.	Matrix Theory	Knowledge of matrix theory to perform row operations on matrices and find bases and dimension of vector spaces and analyze whether a system is consistent or inconsistent and its solution is unique or infinite
3.	Calculus	Proficiency in Calculus to mapping known as a linear map is one that maintains the addition and multiplication of vectors throughout the process.
4.	Geometry	Basic understanding of programming concepts ,as Verilog is a hardware description language (HDL) used for digitaldesign
5.	Group Theory	Familiar with real and complex numbers, and understand the concept of functions
6.	Fundamental Mathematics Knowledge	Knowledge of addition, multiplications, matrix multiplications. Knowledgeof Representasystemintheformoflinearequations.
7.	Previous Coursework	CompletionofintroductorycoursesinLinear Algebraorarelatedfield

2. Competencies

S/L	Competency	KSA Description
1.	**Vector Space**	**Knowledge:** Vector Spaces, Subspaces, linearly dependence and linearly independence **Skills:** Identify vector spaces and subspaces **Attitudes:** Appreciationforperform row operations on matrices and find bases and dimension of vector spaces.
2.	**Basis and Dimension**	**Knowledge:** Basis and Dimension, Dual Space, Matric Theory **Skills:** Identify the basis and dimension of a vector space **Attitudes:** Appreciationforperform row operations on matrices and find basis and dimension of vector spaces.
3.	**Linear Transformations**	**Knowledge:** Algebra of Linear Transformation, Inverse of a linear transformation **Skills:** Apply Linearly transform the system from one dimension to another and represent the pertinent linear transformation in matrix form. **Attitudes:** Linearlytransformthesystemfromonedimensiontoanotherand representthepertinent linear transformation in matrix form.
4.	**Denumerable and non-Denumerable sets**	**Knowledge:** Open set and closed set. Heine-Borel Theorem for closed and bounded intervals. Bolzano Weirstrass Theorem **Skills:** Provides a useful tool to find a measurable subset of Euclidean space **Attitudes:** Appreciationfor finding the measurable subset
5.	**Metric Space**	**Knowledge:** Interior, closure and boundary of a set-in metric spaces **Skills:** Finding the similarity measure and distance-based algorithms. **Attitudes:** AppreciationforAnalyze similarity measure and distance-based algorithms

3. Syllabus

Vector Space and Metric Space **SEMESTER – IV**			
Course Code	**M23BCS407D**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours**	Total Marks	**100**

Credits	**03**	Exam Hours	**03**

Course objectives: This course will enable students to:

1. Identify vector spaces and subspaces.
2. Find the solution of the system of linear equations using matrix operations.
3. Transform a vector space of one dimension into another.
4. Factorize a given matrix using different methods.
5. Identify the metric system and interpret the example of metric space

Module -1

Definition and examples of vector spaces, subspaces of a vector space and the quotient space, Linearly dependence and linearly independence of a set of vectors, Linear span, Exercises and examples based on these concepts.

Module -2

Basis and dimension of a vector space, isomorphic vector spaces, finite and infinite dimensional vector spaces with plenty of examples, Dual space of a finite dimensional vector space-definition and examples, Dimension of dual space, Exercises based on these concepts

Module -3

Linear Transformations on vector space and their examples, algebra of linear transformations on a vector space, matrix representation of a linear trans- formation on finite dimensional vector spaces. Kernel and range of a linear transformation, inverse of linear transformation on finite dimensional vector spaces. Examples and exercises based on these concepts

Module -4

Denumerable and non-Denumerable sets and their examples. Open set and closed set on the real line, their examples and properties. Limit point of a setHeine-Borel Theorem for closed and bounded intervals. Bolzano Weirstrass Theorem. Examples and exercises based on these concepts.

Module -5

Definition of metric space with examples. Open sets and closed sets in metric space. Interior, closure and boundary of a set in metric spaces. Equivalent conditions for open sets and closed sets. Convergence of sequences. Continuous maps and their characterizations. Examples and exercises based on these concepts.

Text Books:

1. Kenneth Hoffman, Ray Kunze, Linear Algebra, Prentice Hall India.
2. N.S. Gopalakrishnan, University Algebra, New Age International (P) Limited, Publishers.

ReferenceBooks:

1.Singh, S. and Zameerudin, Q.,Modern Algebra, Vikas Publishing House Pvt. ltd.
2. Shanti Naryanan, M. D. Rai Singhania, Elements of Real Analysis, S. Chand Publishing House Pvt. Ltd

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Vector space	Definition and examples of vector spaces, subspaces of a vector space Worked Problems Quotient space and Problems Linearly dependence and linearly independence of a set of vectors Worked Problems Linear span, Exercises and examples based on these concepts.
2	Week 3-4: Basis and Dimension	Basis and dimension of a vector space, isomorphic vector spaces finite and infinite dimensional vector spaces with plenty of examples Worked Problems Dual space of a finite dimensional vector space Definition and examples, Dimension of dual space
3	Week 5-6: Linear Transformations:	Linear Transformations on vector space and their examples Algebra of linear transformations on a vectorspace, Worked Problems Matrix representation of a linear trans-formation on finite dimensional vectorspaces. Kernel and range of a linear transformation, inverse of linear transformation on finite dimensional vector spaces
4	Week 7-8: Denumerable and non-Denumerable sets	Denumerable and non-Denumerable sets and their examples Open set and closed set on the real line, their examples and properties Limit point of a set Heine-Borel Theorem for closed and bounded intervals Bolzano Weirstrass Theorem.

		Examples and exercises based on these concepts
5	Week 9-10: Metric Space	Definition of metric space with examples Open sets and closed sets in metric space. Interior, closure and boundary of a set in metric spaces. Equivalent conditions for open sets and closed sets Convergence of sequences. Continuous maps and their characterizations Examples and exercises based on these concepts.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of theconcepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Engineering Science Course (ES)

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	Total Marks			**50**	**20**

Final CIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

Semester-End Examination

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 question from each module, each of the two questions under a module (with a maximum of 3 sub questions), may have mix of topics under that module if necessary.
3. The students have to answer 5 full questions selecting one full question from each module.
4. Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding Vector Space and Matrices	Students will learn competence with the basic ideas of linear algebra including concepts of linear systems, independence, theory of matrices, linear transformations, bases and dimension, eigenvalues, eigenvectors and

		diagonalization.
2	Designing and Analyze the Mathematical Statement	StudentswilllearntoAnalyzewhetherasystemisconsistentorinconsistentanditssolutionisuniqueorinfinite. Designing logical progressions of precise mathematical statements to justify and communicate your reasoning
3	Proficiency in Matrix	Students will become proficient in writing a matrix and Perform row operations on matrices and find bases and dimension of vector spaces
4	Project-BasedLearning	Through hands-on projects, students will apply their knowledge in data science and data analysis for similarity measure and distanced based algorithm. Clustering algorithms such as k-means and hierarchical clustering utilize distance metric to group similar data points together
5	Collaboration and Communication Skills	Students will work collaboratively in teams on design projects, enhancing their ability to communicate effectively, share ideas, and solve problems collectively.
6	Ethical and Professional Responsibility	Students will understand the ethical and professional responsibilities associated with digital design, including respecting intellectual property rights, ensuring design reliability and security, and adhering to industry standards and best practices.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS407D.1	Apply the concepts of vector space, metric space and linear transformation to solve the engineering problem
M23BCS407D.2	Demonstrate the importance of vector space, basis and metric space in Computer Science Engineering
M23BCS407D.3	Analyze the Computer Science engineering applications through linear transformation and metric space

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS407D.1	**3**												**3**	
M23BCS407D.2		**3**												**3**
M23BCS407D.3	**3**												**3**	
M23BCS407D	**3**	**3**											**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**2**	**5**	**3**			**10**
Module 2	**2**	**5**	**3**			**10**
Module 3	**2**	**5**	**3**			**10**
Module 4	**2**	**5**	**3**			**10**
Module 5	**2**	**5**	**3**			**10**
Total	**10**	**25**	**15**			**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**4**	**10**	**6**			**20**
Module 2	**4**	**10**	**6**			**20**
Module 3	**4**	**10**	**6**			**20**
Module 4	**4**	**10**	**6**			**20**
Module 5	**4**	**10**	**6**			**20**
Total	**20**	**50**	**30**			**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

The "Vector Space and Metric Space" course in the fourth semester of the B.E program lays a strong foundation for several future courses in the undergraduate program. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of digital systems. Here are some notable contributions:

Data classification in Internet of Things: The amount of information increases explosively in Internet of Things, because more and more data are sensed by large amount of sensors. The explosive growth of information makes it difficult to access information efficiently, so it is an effective method to decrease the amount of information to be transferred on network by text classification. The text classification algorithm based on vector space model. This algorithm improves the feature selection and weighting methods by introducing synonym replacement to traditional text classification algorithms. The experimental results show that the proposed classification algorithm has considerably improved the precision and recall of classification

Fuzzy Metric Space: After probabilistic metric space, fuzziness and presented an innovative theory, called fuzzy metric space. The idea of probabilistic metric space has been expanded to fuzzy metric space. The purpose of introducing the fuzzy metric space was to allocate the non-negative fuzzy number to the distance among two points, which is the most appropriate way to describe the fuzzy metric space.

Optimization and Approximation Theory: A wide range of metric spaces offers a powerful method to learn the optimization and approximation theory, variational inequalities, and several other problems

Computer Graphics: Linear transformations are widely used in various fields such as computer graphics, where they are essential for rendering 3D objects onto a 2D screen. They are also crucial in solving systems of linear equations, which is a fundamental problem in numerous scientific and engineering discipline

4th Semester	**Ability Enhancement Course (AE)** **SCALA**	**M23BCS408A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Understandingoffundamentalprogrammingconceptssuchas variables, data types, control structures (if statements, loops), functions, and basic algorithms is crucial.
2	**Object- Oriented Programming (OOP)**	Supports both functional and object-oriented programming paradigms. While not strictly required, familiarity with OOP conceptslikeclasses, objects, inheritance, polymorphism, and encapsulation.
3	**Functional Programming Concepts**	Understandingfunctionalprogrammingconceptssuchashigher- order functions, immutability, pattern matching, and recursion.
4	**Development Environment Setup**	SettingupadevelopmentenvironmentforScalaprogrammingis essential. IDEs such as IntelliJ IDEA, Eclipse, or Visual Studio Code with Scala plugins provide excellent support for Scala development.

2. Competencies

S/L	Competency	KSADescription
1	**Conciseness**	**Knowledge:** Scala allows developers to write concise code, reducing boilerplate and enhancing productivity. **Skills:** Learningthebasicsyntaxofscala. **Attitutde:** Writing code that is as brief and expressive as possible while maintaining clarity and readability.
2	**Functional Programming**	**Knowledge:** Scala provides strong support for functional programming constructs such as pattern matching, enabling developerstowrite clean and expressive code. **Skills:** Practicing the concept of functional programming **Attitude:** Scala is a mindset that emphasizes certain principles, such as immutability, higher-order functions, and declarative style, over the imperative style typically seen in object-oriented programming.
3	**Object-Oriented Programming**	**Knowledge:** Scalaisfully object-oriented, supporting features such as classes, inheritance, and polymorphism. It integrates with existing Java libraries and frameworks. **Skills:** Understanding the concept of OOP concepts and libraries. **Attitude:** Scala revolves around organizing and structuring code into objects, classes, and traits, with a focus on encapsulation, inheritance, polymorphism, and abstraction
4	**Concurrency:**	**Knowledge:** Scalaprovidespowerfulconcurrencyprimitivessuchasactors and futures, making it well-suited for building scalable and concurrent applications. **Skills:** Learning relevant applications of scala. **Attitude:** Scala is built around the idea of handling multiple tasks at the same time in a safe, efficient, and scalable manner.
5	**Tooling and Ecosystem**	**Knowledge**: Scala has a rich ecosystem of libraries and frameworks for various use cases, including web development, data processing, and concurrency **Skills:** Applying programming skills to build a new application. **Attitude**: Scala emphasizes providing developers with powerful, flexible, and integrated tools that support both productivity and performance in the development of Scala applications.

3. Syllabus

Dean Academics
MIT Mysore

APPROVED
MAHARAJA INSTITUTE OF TECHNOLOGY
MIT
MYSORE
Dean Academics

Principal
MIT Mysore

SCALA
SEMESTER– IV

CourseCode	**M23BCS408A**	CIEMarks	**50**
NumberofLectureHours/Week (L:T:P: S)	**(0:0:2:0)**	SEE Marks	**50**
TotalNumberofLectureHours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**2**

Course objectives: This course will enable students to:

- Understand the concepts of object-oriented principles with the basic ideas of scala.
- Formulating the solution to complex problems with concepts of decision making, Looping and Inheritance
- Designing the solution to the problems with the knowledge of List, tuple and File handling.
- AnalyzingthecomplexproblemswiththeFunctionalconcept.
- Applythenewemerging tooltomodelthesolutiontovariousproblemsusing scala

PRACTICALCOMPONENT

.1	Assumethefollowinglexicalcouponcodes:**"A","BB","CCC","DDDD","EEEEE"**. Write a Scala program to create a new set of coupon codes based on the above one. Theformatforeachcouponcodeshouldbeasfollows:**"couponcode-i",** where the number i is derived from the length of each corresponding coupon code.
2	CreateaScalaprogramtocalculatethetotalcostforacustomerwhoisbuying10Glazeddonuts where price of each Glazed donut item is at **$2.50.** The format of the $25.00 total cost literal, which is essentially at 2 decimal places.
3	Create a Scala programto prompt customers fortheir name and age. The format forthe name and age labels should be in bold and the name literalshould be underlined.
4	Createa Scalaprogram thatdefines asequenceof numbersfrom 100to110. Thesequence should include the 100 starting number literals, and ends with the 110 number literal.
5	Create aScala programand use an appropriate data structureto present the following keyand value pairs ofchildrenand their ages: **Billis 9 years old, Jonnyis 8years old, Tommy is 11 years old, and Cindy is 13 years old**. Sort out the corresponding child to age in reverse alphabet order.
6	Assume that you two shopping baskets with a bunch of items in each one. The first contains elements:**"Cake","Milk","Cheese",** and the secondonehasthe following items:**"Bread", "Juice", "Milk", "Cheese"**. Write a Scala program to find the common items between the two shopping baskets
7	Assume two data structuresto representthe following numerals: (1) **1,3,5,10, 20**and(2)**20, 17,18,99,0**.WriteaScalaprogramtofindthenumberliteralsthatareinthefirstdatastructure, and second data structure.
8.	WriteaScalaprogramanduseanappropriatedatastructuretorepresentthefollowing number literal:**99.5,100.0,50.0,55.0,70.0,100.0,-1.0**.Assumeaseconddatastructurewiththe followingnumberliterals:**10.0,20.0,30.0,40.0,50.0**. Join the two data structures to gether and find the lowest and the largest number literal from the combined number literals.
9.	WriteaScalaprogramanduseaSequencedatastructureto storeacombinationofnamesto ages as follows: **Seq("James", 7,"Andy", 8,"Tommy", 10,"Bob", 13,"Sam", 10)**, From the above Sequence, extract the age number literal, and use this to find the sumofages.
10.	Write a Scala program and use the Tuple type to represent a shopping cart item with the following properties: **a name, a price, and a quantity bought**. Use a case class to represent the above shopping cart item.

TextBooks:

1.Scala For Beginners, Nadim Bahadoor

2.Introduction to Programming and Problem-Solving Using Scala, Mark C. Lewis,Lisa L Lacher.

References:

1.Begining Scala Vishal Layka and David Pollak,2nd Edition

2.Functional Programming using Scala,Paul Chiusano,Runar Bjarnason

http://leanpub.com/scalaforbeginnershttps://ww

w.javatpoint.com/simple-program-of-scala
https://www.geeksforgeeks.org/scala-tutorial-learn-scala-with-step-by-step-guide/

4. SyllabusTimeline

S/ L	Syllabus Timeline	Description
1	Week 1-10: All lab Experiments	**Competency**: Scala programming **Knowledge:** Importance of Scala programming, abilityto address the challenges of modern software development, including scalability, concurrency, maintainability, and productivity. **Skills:** Building Scalable, efficient, and maintainable applications across various domains, including web development, big data processing, distributed systems using IntelliJ.
2	Week11-12: Exploringon IntelliJ Tool and Lab Internals, Assessment Evaluation	**Competency**: Understanding oftoolusage **Knowledge:** Obtaining more information and knowledge on the tool usage. **Skills:** Analysing the problem to obtain the solution real time problems.

5. Teaching-LearningProcessStrategies

S/L	TLPStrategies:	Description
1	**LectureMethod**	Utilize various teaching methods within the lecture format to reinforce competencies.
2	**Video/Animation**	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**HigherOrder Thinking (HOTS) Questions**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning(PBL)**	Implement PBL to enhance analytical skills and practical application of competencies.
6	**Pair Programming**	Incorporate pair programming sessions where students collaborate in pairs to solve coding tasks or work on projects together.
7	**CaseStudiesand Best Practices**	Analyzing code snippets, architectural decisions, and design patterns employed in these projects to help students understand how Scala is applied in practice.
8	**Problem-Solving Sessions**	Organize problem-solving sessions where students can work together to solve coding challenges and overcome programming obstacles.

6. Assessment Details (both CIE and SEE)

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. LearningObjectives

S/L	Learning Objectives	Description
1	**Understanding Syntax and BasicConcepts:**	The fundamental syntax of Scala, including variables, data types, control structures, functions, and basicobject-orientedconceptslike classes and objects.
2	**Functional Librariesand Frameworks**	ExploringpopularfunctionallibrariesandframeworksintheScala ecosystem, suchasCats, Scala, and fs2, andunderstandingto use themto build functional, composable, and type-safe applications..
3	**Real-World Applications:**	ApplyScalaknowledgetoreal-worldprojectsorproblem-solving exercises to gain practical experience and reinforce learning.
4	**Interoperability with Java**	Understanding the Scala interoperates with Java and learning best practicesforintegratingScalacodewithexistingJavalibrariesand frameworks.

8. CourseOutcomes(COs)andMappingwithPOs/PSOs

Course Outcomes (COs)

COs	Description
M23BCS408A.1	Understand the concepts of object-oriented principles with the basic ideas of scala.
M23BCS408A.2	Formulating the solution to complex problems with concepts of decision making, Looping and Inheritance
M23BCS408A.3	Designing the solution to the problems with the knowledge of List, tuple and File handling.
M23BCS408A.4	Analyzing the complex problems with the Functional concept.
M23BCS408A.5	Apply the new emerging tool to model the solution to various problems using scala

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO11	PO12	PSO 1	PSO 2
M23BCS408A.1	3													
M23BCS408A.2		3										3		
M23BCS408A.3			3									3	3	3
M23BCS408A.4				3									3	3
M23BCS408A.5					3							3	3	3
M23BCS408A	3	3	3	3	3							3	3	3

9. AssessmentPlan

Continuous Internal Evaluation(CIE)

	CO1	CO2	CO3	CO4	CO5	Total
Program 1 to 10	**10**	**10**	**10**	**10**	**10**	**50**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	Total
Program 1 to 10	**10**	**20**	**20**	**20**	**30**	**100**
Total	**10**	**20**	**20**	**20**	**30**	**100**

ConditionsforSEEPaperSetting:

EachmoduleofSEEquestionpaper should beallocatedwithquestions for 20% of thetotal SEE marks

10. Future with this Subject

Scala's future appears bright, driven by its strong features, ecosystem growth, community support, and its relevance in emerging trends such as functional programming, big data, microservices, and type-level programming. As organizations increasingly prioritize scalability, concurrency, and developer productivity, Scala is likely to remain a valuable tool for building robust and scalable software systems.

1. **Functional Programming Adoption**: Functionalprogrammingparadigmsaregaining popularity due to their suitability for building scalable, concurrent, and maintainable systems. Scala'sstrongsupportforfunctionalprogrammingmakesitwell-positionedto capitalize on this trend.
2. **Data Science and BigData**: Scalaisincreasinglybeingusedinthefieldofdatascience and big data processing. Frameworks like Apache Spark, which are written in Scala, have propelled its adoption in this domain. As the demand for data-driven insights continues to grow, Scala's role indata science and big data is likelyto expand further.
3. **Micro services Architecture**: Scala's support for building scalable and concurrent applications makes it well-suited for microservices architecture. As organizations increasinglyadoptmicroservicestobuildmodular, flexible, and scalable systems, Scala is expected to play a significant role in this ecosystem.
4. **Type-Level and Metaprogramming**: Scala's type system is one of its distinguishing features, and there is ongoing research and development in the area of type-level programming and metaprogramming. Advanced type-level featuresenable developers toexpresscomplexconstraintsandenforcecorrectnessatcompiletime, leadingtomore robust and scalable systems.
5. **Tooling and Ecosystem Growth**: The Scala ecosystemcontinues to evolve, with the emergence of new libraries, frameworks, and tools aimed at improving developer productivity and scalability. Improved tooling, IDE support, and build systems contribute to making Scala development more accessible and efficient.

6. **Integration with Other Technologies**: Scala’sinteroperabilitywithJavaanditsability toleverageexistingJavalibrariesandframeworksmakeitanattractivechoicefor organizationswithJava-based infrastructures. As organizations seek to modernize their systems and adopt new technologies, Scala's compatibility with Java provides a smooth transition path.
7. **Community Engagement and Support**: The Scala community is vibrant and active, with ongoing contributions from developers, organizations, and academia. Community- driven initiatives, such as conferences, meetups, and online forums, foster collaboration, knowledge sharing, and the advancement of the Scala ecosystem.

4th Semester	Ability Enhancement Course (AE) MongoDB	M23BCS408B

1. Prerequisites

SL	Proficiency	Prerequisites
1	Install and configure MongoDB locally.	Basic knowledge of operating systems and command line interfaces.
2	Connect to a MongoDB instance using the MongoDB shell	Familiarity with command line interfaces.
3	Perform CRUD operations (Create, Read, Update, Delete) on MongoDB documents.	Basic understanding of database concepts (collections, documents).
4	Design and execute basic queries using MongoDB's query language.	Familiarity with query syntax and operators.
5	Use regular expressions to perform complex pattern matching in queries.	Basic knowledge of regular expression syntax.
6	Construct and execute aggregation pipelines to aggregate, group, and transform data.	Understanding of aggregation concepts (grouping, averaging, etc.).
7	Create and utilize indexes to optimize query performance.	Knowledge of how indexes work and their benefits.
8	Define schema and validation rules to ensure data integrity	Familiarity with data validation principles.
9	Deploy a MongoDB database on a cloud platform (MongoDB Atlas).	Basic knowledge of cloud computing concepts.
10	Create users and assign roles with different privileges.	Understanding of user management and access control.
11	Connect to MongoDB from Python using the pymongo library.	Basic Python programming skills.
12	Perform CRUD operations and execute queries from Python.	Experience with database interactions in Python.
13	Configure security settings for a MongoDB cluster on a cloud platform.	Understanding of network security and access control.
14	Develop serverless functions that interact with a MongoDB Atlas cluster.	Basic knowledge of serverless computing frameworks (AWS Lambda, Google Cloud Functions).

2. Competencies

	Competency	Description (KSA)
1	MongoDB Installation and Configuration	**Knowledge:** Understanding of MongoDB's architecture and installation processes. **Skills:** Ability to install and configure MongoDB on different operating systems. **Attitudes:** Attention to detail, problem-solving, and documentation.
2	MongoDB Shell Proficiency	**Knowledge:** Understanding of MongoDB Shell commands and syntax. **Skills:** Ability to interact with MongoDB databases using the Shell, including performing CRUD operations and executing queries.

		Attitudes: Curiosity, logical thinking, and precision.
3	Data Modeling and CRUD Operations	**Knowledge:** Understanding of document-based data modeling principles. **Skills:** Ability to create, read, update, and delete documents in MongoDB collections. **Attitudes**: Organization, structured thinking, and attention to detail.
4	Advanced Querying Techniques	**Knowledge:** Understanding of advanced MongoDB query operators, including regular expressions and aggregation framework. **Skills:** Ability to design and execute complex queries to extract specific data and perform analysis. **Attitudes:** Problem-solving, critical thinking, and innovation.
5	Performance Optimization and Indexing	**Knowledge:** Understanding of indexing concepts and their impact on query performance. **Skills:** Ability to create and utilize indexes to improve data retrieval speed. **Attitudes:** Analytical thinking, optimization, and efficiency.
6	Data Validation and Integrity	**Knowledge:** Understanding of data validation principles and techniques. **Skills:** Ability to define schema, validation rules, and implement data integrity checks in MongoDB. **Attitudes:** Attention to detail, accuracy, and adherence to best practices.
7	Cloud Deployment with MongoDB Atlas	**Knowledge:** Understanding of cloud computing concepts and MongoDB Atlas features. **Skills:** Ability to create and configure MongoDB Atlas clusters, manage access, and perform operations in a cloud environment. **Attitudes:** Adaptability, exploration, and technological awareness.
8	User Management and Access Control	**Knowledge:** Understanding of user roles, permissions, and security protocols. **Skills:** Ability to create users, assign roles, and manage access control for MongoDB databases. **Attitudes:** Security-mindedness, responsibility, and adherence to security best practices.
9	MongoDB Integration with Python	**Knowledge:** Understanding of MongoDB's Python driver (pymongo). **Skills:** Ability to connect to MongoDB from Python, perform CRUD operations, and execute queries using the pymongo library. **Attitudes:** Problem-solving, resourcefulness, and adaptability.
10	Cloud Integration and Serverless Development	**Knowledge:** Understanding of serverless computing concepts and cloud service platforms (AWS Lambda, Google Cloud Functions). **Skills:** Ability to develop serverless functions that interact with MongoDB databases in a cloud environment. **Attitudes:** Innovation, adaptability, and a desire to learn new technologies.

3. Syllabus

MongoDB		Semester	**4**
Course Code	**M23BCS408B**	CIE Marks	**50**
Teaching Hours/Week (L: T:P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Hours of Pedagogy	**24 Hours**	Total Marks	**100**
Credits	**01**		
Sl.NO	**Experiments**		

Dean Academics MIT Mysore

APPROVED MYSORE Dean Academics

Principal MIT Mysore

Lab 1:

Aim: Introduction to NoSQL and MongoDB

1a. Install MongoDB on your system (or use a pre-configured instance). Connect to the database using the MongoDB shell. Verify your connection by listing all the databases.

1b. Create a new database named 'my_database'. Switch to this database and create a collection named 'students'. Insert two sample documents into the 'students' collection, each representing a student with the following fields: 'name', 'age', 'city'.

Lab 2:

Aim: Data Modelling and CRUD Operations.

2a. Add a new field 'email' to the existing student documents in the 'students' collection. Update the documents to include email addresses for each student.

2b: Write a MongoDB query to retrieve all student documents with an age greater than 20. Additionally, write another query to retrieve students whose city is 'Bengaluru'.Master CRUD operations (Create, Read, Update, Delete) on MongoDB documents.

Lab 3:

Aim: Advanced Queries and Regular Expressions

3a. Use the '$regex' operator to retrieve all student documents whose name starts with tahe letter 'A'. Then, retrieve students whose city contains the word "Delhi".

3b. Create a new collection named 'books' and populate it with sample documents representing different books, each containing fields like 'title', 'author', and 'genre'. Write a query to retrieve all books that have the genre 'Science Fiction'.

Lab 4:

Aim: Aggregation and Data Analysis

4a. Write an aggregation pipeline to calculate the average age of students in the 'students' collection.

4b. Use an aggregation pipeline to group students by city and count the number of students in each city. Then, sort the results in descending order based on the count.

Lab 5:

Aim: Indexing and Performance Optimization

5a. Create an index on the 'age' field in the 'students' collection. Execute a query to retrieve students with an age greater than 25 and compare its execution time before and after creating the index.

5b. Create a compound index on 'city' and 'age' fields. Run a query to retrieve students from 'Mysuru' who are older than 20 and analyze the improvement in performance compared to a query without an index.

Lab 6:

Aim: Data Validation and Integrity

6a. Define a schema for the 'students' collection, specifying data types and required fields. Add a custom validation rule to ensure that the email field is a valid email format.

6b. Attempt to insert a document with invalid data (e.g., an invalid email address or missing required fields) and observe the error message. Explain the error message and how it relates to the defined schema and validation rules.

Lab 7:

Aim: Creating clusters using MongoDB Atlas

7a. Create a free MongoDB Atlas cluster and connect to it using the MongoDB Shell.

7b. Create a new database and collection within your Atlas cluster. Insert a few sample documents into the collection and verify that data is being saved to the cloud instance.

Lab 8:

Aim: User Management and Access Control

8a. Create a new user with read-only access to the 'students' database. Try to perform write operations using this user and analyze the results.

8b. Create a role with read and write access to the 'books' collection and assign this role to a new user. Verify that the user has the expected permissions.

Lab 9:

Aim: MongoDB and Python

9a. Using the 'pymongo' library, connect to a local MongoDB instance. Retrieve all documents from the 'students' collection and print them to the console.

9b. Write a Python script to insert a new document into the 'students' collection using the pymongo library. Then, retrieve the newly inserted document and print it.

Lab 10: MongoDB and Cloud Integration

10a. Set up a MongoDB Atlas cluster with a suitable cloud provider (AWS, Azure, GCP). Configure security settings to restrict access to the cluster based on IP addresses.

10b. Use a cloud service like AWS Lambda or Google Cloud Functions to create a serverless function that interacts with your MongoDB Atlas cluster. This function should perform a simple task like inserting a new document into a collection or retrieving data based on a specific query. The data could represent information about businesses in different Indian cities.

Suggested Learning Resources:

BOOK1:“MongoDB:TheDefinitiveGuide”,Kristinachodorow,2ndedO’REILLY,2013.

BOOK2:“*MongoDBinAction*”byKYLEBANKERet.al.2nded,Manningpublication,2016

BOOK3:“MongoDBCompleteGuide”byManuSharma1sted,bpbpublication,2023.

Online resources

- **installationofMongoDBVideo:**https://www.youtube.com/watch?v=dEm2AS5amyA
- **videoonAggregation:**https://www.youtube.com/watch?v=vx1C8EyTa7Y
- **MongoDBinactionbookCodedownloadURL:h**ttps://www.manning.com/downloads/529
- **MongoDBExerciseURL:**https://www.w3resource.com/mongodb-exercises/

4. Syllabus Timeline

Sl No	Program	Competency (KSA) Developed
1	Lab 1: NoSQL & MongoDB Intro (Installation, Connection)	Install & Configure: Knowledge of MongoDB architecture, **Skills:** Installation, Connection via Shell. **Attitudes:** Problem Solving, Documentation
2	Lab 2: Data Modeling & CRUD (Document Manipulation)	CRUD Mastery: Knowledge of Data Modeling, **Skills:** Create, Read, Update, Delete. **Attitudes:** Organization, Attention to Detail
3	Lab 3: Advanced Queries (Regex, Queries)	Query Expert: Knowledge of Regex, Advanced Querying, **Skills:** Complex Queries, Regex Use. **Attitudes**: Critical Thinking, Innovation
4	Lab 4: Data Aggregation (Aggregation, Grouping)	Data Analysis: Knowledge of Aggregation Framework, **Skills:** Group, Sum, Average Data. **Attitudes:** Analytical Thinking
5	Lab 5: Performance Optimization (Indexing, Query Speed)	Index Mastery: Knowledge of Indexes, **Skills:** Creating & Utilizing Indexes, Performance Comparison. **Attitudes:** Efficiency, Optimization
6	Lab 6: Data Integrity (Schema, Validation)	Data Integrity: Knowledge of Schema Design, Validation Rules, **Skills:** Implementing Validation, Error Handling. **Attitudes:** Accuracy, Best Practices
7	Lab 7: MongoDB Atlas (Cloud Deployment)	Cloud Deployment: Knowledge of MongoDB Atlas, **Skills**: Creating & Configuring Atlas Clusters, Data Insertion. **Attitudes**: Adaptability, Exploration
8	Lab 8: User Management (Access Control, Security)	Security Expert: Knowledge of User Roles, Permissions, **Skills:** User Creation, Role Assignment. **Attitudes:** Security-Mindedness, Responsibility
9	Lab 9: Python Integration (Python & MongoDB)	Python & MongoDB: Knowledge of pymongo, **Skills:** CRUD Operations, Queries from Python. **Attitudes:** Problem Solving, Resourcefulness
10	Lab 10: Cloud Integration (Serverless, Cloud Functions)	Serverless Development: Knowledge of Serverless Functions, **Skills:** Developing Functions, Cloud Platform Integration. **Attitudes:** Innovation, Adaptability
11	Lab 11: Project - Data Analysis (Project-based)	Project Presentation and Review
12	Lab 12: Evaluation	Internals

5. Teaching-Learning Process Strategies

S/N	TLP Strategie	Description
1	Interactive Demonstrations	The instructor demonstrates key MongoDB concepts and commands using the MongoDB Shell. Students follow along and can ask questions during the demonstration. This helps build a foundation of understanding and allows for immediate clarification.
2	Hands-on Exercises	Students complete practical exercises based on the lab objectives. These exercises can range from basic CRUD operations to more complex queries and aggregation pipelines. This approach provides active learning and reinforces the concepts learned.
3	Collaborative Learning	Students work in pairs or small groups to solve problems and complete exercises. This encourages peer-to-peer learning, problem-solving, and teamwork.
4	Case Studies and Real-World Examples	The instructor presents real-world scenarios where MongoDB is used. This helps students understand the practical applications of MongoDB and how it solves real-world problems.
5	Code Review and Debugging Sessions	The instructor reviews student code, identifying common errors and providing feedback on best practices. Students can also help each other debug their code, fostering problem-solving skills.
6	Interactive Quizzes and Assessments	Regular quizzes can assess students' understanding of key concepts. Short quizzes can be conducted during or at the end of each lab session to reinforce learning.
7	Project-Based Learning	Students work on a larger project that requires them to apply the learned MongoDB concepts and skills. This approach encourages deeper learning, creativity, and problem-solving.
8	Use of Online Resources and Documentation	Encourage students to utilize official MongoDB documentation, online tutorials, and community forums to further their understanding and explore solutions to problems.
9	Guest Lectures or Industry Experts	Invite professionals from the industry to share their experiences working with MongoDB, providing insights into real-world applications and industry trends.
10	Gamification and Learning Games	Incorporate fun and engaging elements, like quizzes, coding challenges, or online simulations, to make learning more interactive and enjoyable.

6. Assessment Details (both CIE and SEE)

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
	Total	**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
	Total	**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20

Total	**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4.Duration of SEE shall be 3 hours.

7. Learning Objectives

	Learning Objectives	Description
1	Understand the fundamental concepts of NoSQL databases and MongoDB's architecture	This objective sets the stage for the course, introducing students to the core principles of NoSQL databases and how MongoDB operates. It provides context for the upcoming lab programs.
2	Gain hands-on experience with basic CRUD operations and data modeling in MongoDB.	This objective emphasizes the practical skills students will acquire in the initial lab programs. It prepares them for more complex tasks and data analysis in later labs.
3	Develop proficiency in designing and executing advanced queries using MongoDB's query language.	This objective highlights the importance of learning advanced querying techniques, including regular expressions, aggregation pipelines, and indexing. These skills are essential for extracting valuable insights from data.
4	Learn how to optimize query performance and ensure data integrity using indexing and data validation.	This objective underscores the practical aspects of working with large datasets in MongoDB. Students will learn to optimize data access and maintain data quality.
5	Explore the benefits and techniques of deploying MongoDB in the cloud, including user management and security.	This objective aims to introduce students to the real-world applications of MongoDB in cloud environments. They will learn how to secure their data and manage access to cloud-based databases.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS408B.1	Create and manage MongoDB databases and collections and perform basic data manipulation operations (CRUD) using the MongoDB Shell.
M23BCS408B.2	Analyze data stored in MongoDB using aggregation pipelines, group data by various fields, and filter results based on specific criteria.
M23BCS408B.3	Design data models for MongoDB collections, ensuring data integrity and consistency through schema validation and custom validation rules.
M23BCS408B.4	Build simple applications using the pymongo library in Python to interact with MongoDB databases. They can connect to both local and cloud-based MongoDB instances.
M23BCS408B.5	Design and implement secure cloud deployments of MongoDB using services like MongoDB Atlas, understanding the concepts of access control, user roles, and network security.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2

M23BCS408B.1	**2**				**3**								**3**	**3**
M23BCS408B.2		**3**			**2**								**2**	**3**
M23BCS408B.3			**3**		**3**								**3**	**3**
M23BCS408B.4				**3**	**3**								**2**	**3**
M23BCS408B.5			**3**		**3**	**2**							**2**	**3**
M23BCS408B	**2**	**3**	**3**	**3**	**2.6**	**2**							**2.4**	**3**

9.Assessment Plan

Continuous Internal Evaluation (CIE)

Programs	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to Program 12	**10**	**10**	**10**	**10**	**10**	**50**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to Program 12	**20**	**20**	**20**	**20**	**20**	**100**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

Learning MongoDB early in engineering provides a significant advantage. It equips students with valuable skills that are in high demand, enhancing theirr future career prospects .

Benefits during Future Engineering Years.

- **Data-Driven Decision Making:** The ability to work with databases like MongoDB is increasingly crucial in engineering. Students can analyze real-world data to gain insights for their projects, research, or design decisions.
- **Data Management for Projects:** MongoDB's flexibility makes it ideal for storing and managing data related to engineering projects.
- **Building Data-Driven Systems:** MongoDB skills equip students to build data-driven systems, dashboards, or visualizations for project reporting and analysis.
- **Software Development Skills:** Learning MongoDB often involves using programming languages like Python. This strengthens their software development skills, which are valuable across various engineering disciplines.
- **Cloud Integration:** The ability to work with MongoDB on cloud platforms like AWS, Azure, or GCP is a valuable skill in modern engineering. It allows students to build scalable and adaptable systems.
- **Personal Projects:** MongoDB makes it easier to build data-driven personal projects, which can enhance students portfolio and demonstrate their technical capabilities.

Placement and Job Benefits.

- **In-Demand Skill:** MongoDB is a highly sought-after skill in today's job market. Many companies are looking for engineers who can manage data effectively.
- **Competitive Edge:**Having MongoDB skills can give students a significant advantage over other candidates during the placement process.
- **Career Growth:** A strong understanding of databases like MongoDB can open doors to roles in data engineering, data analysis, software development, and cloud computing.
- **Higher Salary Potential:** Data-related skills are often associated with higher salaries, making MongoDB a valuable asset for students career.
- **Adaptability:** MongoDB skills are transferable across different industries and domains.

Specific Job Roles.

- **Data Engineer:** Responsible for designing, building, and maintaining data systems using MongoDB.
- **Data Analyst:** Uses MongoDB to analyze data, extract insights, and create reports.
- **Software Engineer:** Incorporates MongoDB into software applications to manage data.
- **Cloud Engineer:** Builds and manages cloud-based applications and infrastructure, often using MongoDB for data storage.

4th Semester	Ability Enhancement Course (AE) UI/UX	M23BCS408C

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Understanding of Design Principles	Understanding color theory, typography, layout, and visual hierarchy.
2.	User-Centered Design Approach	Ability to empathize with users to understand their needs and pain points.
3.	Technical Skills	Understanding of how designs are implemented on the web.
4.	Information Architecture	Understanding how to organize information and create sitemaps.
5.	Analytical and Critical Thinking	Ability to analyze user data and feedback to inform design decisions
6.	Problem-Solving Skills	Strong skills in identifying and solving design-related problems.

2. Competencies

S/L	Competency	KSA Description
1.	**User Journey Mapping**	**Knowledge:** Understand the fundamental of UI/UX **Skills:** Mapping out user flows and journeys to understand how users interact with a product. **Attitudes:**User interaction
2.	**Organizing Information**	**Knowledge:** Understanding the importance of organizing data. **Skills:** Structuring and organizing content in a way that is easy for users to navigate. **Attitudes:**Managing data
3.	**Creating Sitemaps**	**Knowledge:** Understanding the Information Architecture. **Skills:** Designing comprehensive sitemaps that outline the structure of a website or app.. **Attitudes:**Confident in designing UI
4.	**Wireframing**	**Knowledge:** Understanding the structure of the user interface. **Skills:** Developing low-fidelity wireframes to outline the layout and structure of interfaces. **Attitudes:**Comfortable in designing layouts
5.	**Interaction Design**	**Knowledge:** Understanding the user interaction **Skills:** Creating intuitive and aesthetically pleasing user interfaces for various devices. **Attitudes:**Designing of effective user interface.

3. Syllabus

UI/UX
SEMESTER – IV

Course Code	**M23BCS408C**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**01**

Course objectives:

- Understand the importance of the User Interface Design process.
- To make the learners familiar with the importance of requirement, user analysis and different levels of design for a particular project and the techniques to be used.
- To introduce them to Figma tool- a tool for prototyping

Sl. No.	Experiments ***Note:*** *Experiments to be conducted using Figma*
1	Design a Logo for an E-Commerce app.

2	Design an email that showcases a promotional offer of the e-commerce app.
3	Design brochure that showcases different features of the e-commerce app.
4	Create sketches and low-fidelity wire frames and experiment the user testing
5	Create High-Fidelity Mockups & Prototypes from the wireframes.
6	Figma basics: Creating basic responsive elements like buttons, input elements, etc. to understand frames, groups, layout, constraints, texts, vector, color palette, etc.
7	Basic Clickable Prototyping using figma.
8	Create a Design System for an e-commerce app using Grid and Spacing, Typography, Color System, and UI elements like icons, images, buttons, Inputs, Cards, Search Bar, Lists, etc.
9	Reusing the above Design System, compose the Home page, Product Page, and Checkout Page of the e-commerce app.
10	Test your sitemap using Treejack.

Text Books:

1. Don't Make Me Think" by Steve Krug, "The Design of Everyday Things" by Don Norman.
2. UI UX DESIGN AND FIGMA FUNDAMENTALS: Complete Beginner's Guide & Picture Illustrations to get Started with UI & UX Design in FIGMA (includes Wireframes, Prototypes & Exporting)

Reference Books:

1. Russ Unger, Carolyn Chandler, A Project Guide to UX Design: For User Experience Designers in the Field Or in the Making, New Riders; 2nd edition, 2012.
2. UI UX DESIGN AND FIGMA: Complete Guide & Picture Illustrations With Practical Examples for learning & Mastering UI/UX Design in FIGMA (includes Wireframes, Prototypes & Exporting)
3. Everett N. McKay, UI is Communication: How to Design Intuitive, User Centered Interfaces by Focusing on Effective Communication, Morgan Kaufmann; Illustrated edition, 2013.

Web resources

https://figma.com/community/file/1019904134655257869/ui-ux-starter-guide-free-resources

https://screenlane.com/

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-10: All lab Experiments	**Understand what is user interface and importance of user interface:** Apply of user interface design, requirement analysis during design and development. Familiar with Figma tool for creating User Interface
2	Week 11-12: Explore more on Figma Tool and Lab Internals and Evaluation	**Acquire knowledge on different** tools for designing user interface. Students become strong in analysis and design of user interface according the user requirement.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Project-Based Learning	Design courses around real-world projects that students can work on individually or in teams.
2	Design Sprints	Incorporate design sprints to teach students the iterative process of prototyping, testing, and refining designs.
3	Empathy Exercises	Implement activities that help students develop empathy for users, such as user persona creation and journey mapping.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.

5	Team Projects	Promote collaboration through group projects that mimic real-world team dynamics.
6	Peer Reviews	Encourage peer reviews and critiques to develop critical thinking and feedback skills.
7	Guest Speakers	Invite industry professionals to share their experiences and insights.

6. Assessment Details (both CIE and SEE)

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4.Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understanding the importance of User Interface	Toenablethestudentstobecomeawareofdifferent User interfaces and the need of user interface.
2	Apply design thinking	Students are able to design an interface by applying their design thinking.
3	Develop an effective interface	Student will be able to develop effective user interface by using screen based experiences.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS408C.1	Experiment with various visual design aspects.
M23BCS408C.2	Theme the visual look and feel of the user experiences using figma.
M23BCS408C.3	Create effective and compelling screen based experiences.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS408C.1	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**	**2**
M23BCS408C.2	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**-**
M23BCS408C.3	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**1**	**-**
M23BCS408C	**3**	**3**	**2**		**3**							**2**	**1.5**	**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	**Total**
All Experiments	**10**	**15**	**25**	**50**
Total	**10**	**15**	**25**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	**Total**
All Experiments	**20**	**30**	**50**	**100**
Total	**20**	**30**	**50**	**100**

10. Future with this Subject

The "UI/UX" lab course in the fourth semester of the B.E (Computer Science & Engineering Branches) program places an important role for learning in designing the user interface for software applications.

Here are some notable contributions:

- **Artificial Intelligence and Machine Learning**

Personalization: AI can enable more personalized user experiences by analyzing user behavior and preferences. Automated design tools can assist in creating more efficient workflows and rapid prototyping.

- **User Analytics:** Increasing use of data analytics to inform design decisions, improve user engagement, and personalize experiences.

4th Semester	Ability Enhancement Course (AE) Hands on IOT	M23BCS408D

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Electronics Knowledge:	Understanding concepts such as voltage, current, resistance, and how to use basic electronic components like resistors, capacitors, LEDs, and sensors is crucial for working with Arduino and Raspberry Pi.
2	Programming Skills:	Proficiency in programming languages such as C/C++ (for Arduino) and Python (for Raspberry Pi) is essential. You should be comfortable writing and debugging code, as IoT projects involve both hardware and software components.
3	Familiarity with Arduino and Raspberry Pi Platforms:	Before diving into hands-on projects, it's important to have a good understanding of the Arduino and Raspberry Pi platforms, including their capabilities, pin configurations, GPIO (General Purpose Input/Output) usage, and how to set up and configure them.
4	Understanding of IoT Concepts:	Knowledge of IoT concepts such as data acquisition, data transmission protocols (like MQTT or HTTP), data processing, and data visualization is necessary for building effective IoT solutions. You should understand how sensors collect data, how microcontrollers process it, and how it's communicated to the cloud or other devices.
5	Access to Hardware and Tools:	Access to the necessary hardware components such as Arduino boards, Raspberry Pi boards, sensors (e.g., temperature, humidity, motion), actuators (e.g., motors, servos), breadboards, jumper wires, and power supplies. Additionally, having the appropriate tools for soldering, wire stripping, and circuit assembly can be beneficial.

2. Competencies

S/L	Competency	Description
1	IoT Architecture Design**:**	Understand how to architect IoT systems which involves knowledge of sensor selection, data transmission protocols, cloud infrastructure, and edge computing. In an IoT lab, will learn how to design efficient and scalable IoT architectures tailored to specific use cases.
2	Sensor Integration and Data Processing:	Gain the knowledge of collecting data from various sensors deployed in the physical world. Competency in sensor integration includes selecting appropriate sensors, connecting them to microcontrollers or IoT gateways, and processing the data they generate. This competency often involves programming skills in languages like C/C++, Python, or Java.
3	Data Analytics and Visualization:	How to deal with large volumes of data generated by IoT devices, the ability to analyze and derive insights from this data is crucial. Competency in data analytics involves skills in data preprocessing, statistical analysis, machine learning algorithms, and data visualization techniques. In an IoT lab, learn how to extract meaningful insights from IoT data to inform decision-making processes.
4	Cyber security and Privacy:	As IoT devices become more pervasive, ensuring the security and privacy of IoT systems is paramount. Competency in cyber security involves understanding common IoT security threats, implementing encryption and authentication mechanisms, and adhering to best practices for securing IoT deployments. An IoT lab can provide hands-on experience in implementing security measures and conducting vulnerability assessments to protect IoT systems from cyber attacks.
5	Prototyping and Rapid Iteration:	IoT projects often involve rapid prototyping and iterative development cycles to refine product designs and functionality. Competency in prototyping involves skills in hardware prototyping using development boards like Arduino or Raspberry Pi, as well as software prototyping using frameworks like Node-RED or TensorFlow Lite. In an IoT lab, you can learn how to quickly prototype IoT solutions, gather feedback from users, and iterate on designs to improve

		performance and usability.

3. Syllabus

Hands on IOT
SEMESTER – IV

Course Code	**M23BCS408D**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 Hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**2**

Course objectives: This course will enable students to:

- To impart the concepts of IOT terminology, technology and its applications
- To impart the concepts of M2M (machine to machine) with necessary protocols
- To impart Python Scripting Language and Arduino IDE which is used in many IoT devices
- To impart the concepts of Raspberry PI/Arduino platforms, that is widely used in IoT applications
- To impart the concepts of the implementation of web-based services on IoT devices

LIST OF LAB EXPERIMENTS	**BT**
Study the fundamentals of IOT software's and components.	
2. Familiarization with the detailed concepts of Arduino/Raspberry Pi.	
3. Build a circuit and Develop a program to interface LED/Buzzer with Arduino/Raspberry Pi to turn on LED for 1 second after every 2 seconds.	
4. Design the circuit and develop a program for smart agriculture system using smart phone and simulate the circuit with the help of Arduino.	
5. Design the circuit and develop a program for energy management and monitoring system using Smart phone and simulate the circuit with the help of Arduino.	
6. Build a circuit and develop a program for Wireless sensor network based water tank level monitoring system using Arduino.	
7. Design the circuit and develop a program for Google assisted home automation using Raspberry pi.	
8. Build a circuit using Raspberry pi based weather reporting over IOT.	
9. Design the circuit and develop a program for vehicle number plate recognition system using Raspberry Pi.	
10. Design the circuit using Raspberry Pi and develop a program for voice based hot or cold water dispenser system.	

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1	Introduction to IOT software and components.
2	Week 2	Familiarization with the detailed concepts of Arduino /Raspberry Pi.
3	Week 3	Program to interface LED/Buzzer with Arduino
4	Week 4	Program to develop smart agriculture system using smart phone
5	Week 5	Program for Energy management and monitoring system
6	Week 6	Program for Water tank level monitoring system.
7	Week 7	Program for Google Assisted Home Automation.
8	Week 8	To build a circuit for Weather Reporting Over IOT.
9	Week 9	To develop vehicle number plate recognition system.
10	Week 10	To develop hot or cold water dispenser system

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.

3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Class Work:-A

CIE Split up for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 30marks (60% of the maximum Marks)

Laboratory Test: -B

CIE Split up for Test in Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Write-up, Conduction, result and Procedure	60%	30
2	Viva-Voce	40%	20
Total		**100%**	**50**

The Test marks should be scaled down to 20marks (40% of the maximum Marks)

Final CIE for Laboratory based Ability Enhancement Course

SL. No.	Description	% of Marks	In Marks
1	Scaled Down marks of record/journal-A	60% of the maximum	30
2	Scaled Down marks of test-B	40% of the maximum	20
Total		**100%**	**50**

FinalCIE Marks =(A) + (B)

SEE for practical Course:

1. SEE marks for practical course shall be 50 marks

SL. No.	Description	% of Marks	Marks
1	Write-up, Procedure	20%	20
2	Conduction and result	60%	60
3	Viva-Voce	20%	20
Total		**100%**	**100**

2. SEE for practical course is evaluated for 100 marks and scored marks shall be scaled down to 50 marks.
3. Change of experiment/program is allowed only once and 20% marks allotted to the procedure/write-up part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives
1	Understand the basics of IoT (Internet of things) and its applications.
2	Explore different types of sensors and actuators used in IoT applications.
3	Learn how to set up and configure Arduino and Raspberry Pi for IoT projects.
4	Gain Proficiency in programming Arduino and Raspberry Pi to collect sensor data

5	Learn how to establish communication between Arduino/Raspberry Pi and the cloud or other devices.
6	Understand the concept of data visualization and analysis for IoT applications.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BCS408D.1	Understand and apply basics of Iot (Internet of things) and its applications.
M23BCS408D.2	Interface Input/ Output devices, sensors and communication modules.
M23BCS408D.3	Create a solution to perform remotely monitor data and control devices.
M23BCS408D.4	Develop real life IOT based projects.
M23BCS408D.5	Evaluate various IOT protocols for designating real world scenarios.

CO-PO-PSO Mapping

COs/POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO2
M23BCS408D.1	3												3	
M23BCS408D.2		3												3
M23BCS408D.3			3		3								3	
M23BCS408D.4			3		3									3
M23BCS408D.5				3									3	
M23BCS408D	3	3	3	3	3								3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**10**	**10**	**10**	**10**	**10**	**50**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1 to 10	**20**	**20**	**20**	**20**	**20**	**100**
Total	**20**	**20**	**20**	**20**	**20**	**100**

Conditions for SEE Paper Setting:

Each module of SEE question paper should be allocated with questions for 20% of the total SEE marks

10. Future with this Subject

Engaging in hands-on IoT projects offers numerous opportunities and potential impacts on the future, both on an individual level and for society as a whole:

1. Career Opportunities: Hands-on IoT experience equips individuals with highly sought-after skills in electronics, programming, data analytics, and system integration. This can open doors to diverse career paths in fields such as IoT development, embedded systems engineering, data science, and smart infrastructure management.

2. Innovation and Entrepreneurship: IoT projects foster innovation by empowering individuals to develop and prototype novel solutions to real-world problems. This could lead to the creation of new products, services, and business models, driving entrepreneurship and economic growth.

3. Advancements in Technology: As more individuals and communities engage in hands-on IoT projects, we can expect to see accelerated advancements in IoT technologies. This includes improvements in sensor technology, communication protocols, edge computing, and artificial intelligence, leading to more sophisticated and efficient IoT systems.

4. Smart Cities and Sustainable Development: IoT projects have the potential to transform cities into smart, interconnected ecosystems where data-driven decision-making optimizes resource utilization, enhances public services, and improves quality of life. This could contribute to more sustainable and resilient urban development.

5. Empowerment and Inclusivity: Hands-on IoT projects democratize access to technology by providing individuals with the tools and knowledge to create their own IoT solutions. This empowerment fosters inclusivity by enabling participation from diverse communities, including hobbyists, students, entrepreneurs, and citizen scientists.

6. Addressing Societal Challenges: IoT projects can be instrumental in addressing pressing societal challenges such as healthcare access, environmental monitoring, disaster response, and transportation efficiency. By leveraging IoT technology, communities can develop innovative solutions to complex problems, leading to positive social impact.

7. Lifelong Learning and Skill Development: Engaging in hands-on IoT projects promotes lifelong learning and skill development. As technology evolves, individuals must continually update their knowledge and adapt to new tools and techniques. Hands-on projects provide opportunities for ongoing exploration, experimentation, and skill enhancement.

Overall, the future with hands-on IoT projects holds promise for technological advancement, economic growth, social empowerment, and sustainable development. By fostering creativity, collaboration, and innovation, hands-on IoT initiatives have the potential to shape a more connected, efficient, and resilient world.

4th Semester	**Ability Enhancement Course (AE) Universal Human Values (UHV)**	**M23BUHK409**

Universal Human Values (UHV)

Course Code	**M23BUHK409**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**12**	Total Marks	**100**
Credits	**01**	Exam Hours	**01 Hour**

Course objectives:

This course is intended to:

1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and the movement towards value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behaviour and mutually enriching interaction with Nature.
4. This course is intended to provide a much-needed orientation input in value education to the young enquiring minds.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematic andrational study of the human being vis-à-vis the rest of existence.
2. In addition to the traditional lecture method, different types of innovative teaching methods may be adoptedso that the activities will develop students' theoretical and applied skills.
3. State the need for UHV activities and their present relevance in society and provide real-life examples.
4. Support and guide the students in self-study activities.
5. You will also be responsible for assigning homework, grading assignments and quizzes, and documentingstudents' progress in real activities in the field.
6. This process of self-exploration takes the form of a dialogue between the teacher and the students to beginwith, and then to continue within the student in every activity, leading to continuous self-evolution.
7. Encourage the students for group work to improve their creative and analytical skills.

Module-1

Introduction to Value Education (3 hours)

Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations.

Module-2

Harmony in the Human Being (3 hours)

Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health

Module-3

Harmony in the Family and Society (3 hours)

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationships,
'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationships, Understanding Harmony in Society, Vision for the Universal Human Order

Module-4

Harmony in the Nature/Existence:(3 hours)

Understanding Harmony in Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence

Module-5

Implications of the Holistic Understanding – a Look at Professional Ethics:**(3 hours)**

Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics HolisticTechnologies, Production Systems and Management Models-Typical Case Studies, Strategies for Transition towards Value-based Life and Profession

Course outcome (Course Skill Set):

At the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature);

COs	Description
M23BUHK409.1	They would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
M23BUHK409.2	They would have better critical ability.
M23BUHK409.3	They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society).
M23BUHK409.4	It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

Expected to positively impact common graduate attributes like:

1. Ethical human conduct
2. Socially responsible behaviour
3. Holistic vision of life
4. Environmentally responsible work
5. Having Competence and Capabilities for Maintaining Health and Hygiene
6. Appreciation and aspiration for excellence (merit) and gratitude for all

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). The student is declared as a pass in the course if he/she secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous internal Examination (CIE)

1. For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
2. The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
3. Any two assignment methods mentioned in the regulations, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
4. For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.

The sum of two tests, two assignments, will be out of 100 marks and will be scaled down to 50 marks.

Internal Assessment Test question paper is designed to attain the different levels of Bloom’s taxonomy as per the outcome defined for the course.

Semester End Examinations (SEE)

SEE paper shall be set for **50 questions**, each of the 01 marks. **The pattern of the question paper is MCQ (multiple choice questions). The time allotted for SEE is 01 hour.** The student has to secure a minimum of

35% ofthe maximum marks meant for SEE.

Suggested Learning Resources:

Books for READING:

Text Book and Teachers Manual

1. The Textbook A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G PBagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978- 93-87034- 47-1
2. The Teacher‟s Manual for A Foundation Course in Human Values and Professional Ethics, R R Gaur, RAsthana, G

Reference Books

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amar kantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj – Pandit Sunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)
14. Sussan George, 1976, How the Other Half Dies, Penguin Press. Reprinted 1986, 1991
15. Donella H. Meadows, Dennis L. Meadows, Jorgen Randers, William W. Behrens III, 1972, Limits to Growth –

Club of Rome’s report, Universe Books.

16. A Nagraj, 1998, Jeevan Vidya Ek Parichay, Divya Path Sansthan, Amarkantak.
17. P L Dhar, RR Gaur, 1990, Science and Humanism, Commonwealth Publishers.
18. A N Tripathy, 2003, Human Values, New Age International Publishers.
19. SubhasPalekar, 2000, How to practice Natural Farming, Pracheen (Vaidik) KrishiTantraShodh, Amravati.
20. E G Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers , Oxford UniversityPress
21. M Govindrajran, S Natrajan & V.S. Senthil Kumar, Engineering Ethics (including Human Values), EasternEconomy Edition, Prentice Hall of India Ltd.
22. B P Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
23. B L Bajpai, 2004, Indian Ethos and Modern Management, New Royal Book Co., Lucknow. Reprinted 2008.

Web links and Video Lectures (e-Resources):

1. Value Education websites,
2. https://www.uhv.org.in/uhv-ii
3. http://uhv.ac.in
4. ttp://www.uptu.ac.in
5. Story of Stuff.
6. http://www.storyofstuff.com
7. Al Gore, An Inconvenient Truth, Paramount Classics, USA
8. Charlie Chaplin, Modern Times, United Artists, USA
9. IIT Delhi, Modern Technology – the Untold Story
10. Gandhi A., Right Here Right Now, Cyclewala Productions
11. https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEkQw
12. https://fdp-si.aicte-india.org/8dayUHV_download.php
13. https://www.youtube.com/watch?v=8ovkLRYXIjE

Distribution of Activities-Semester wise from 3rdto6th semester

4th Semester	**National Service Scheme /Physical Education/Yoga(NCMC)** **National Service Scheme (NSS)**	**M23BNSK410**

Sem	Topics/ActivitiestobeCovered
3rdSemfor 25 Marks	1. Organic farming, Indian Agriculture(Past,Present,andFuture) Connectivity for marketing. 2. Waste management–Public, Private and Govt organization,5R's. 3. Setting of the information imparting club for women leading to contribution in social and economic issues.
4thSemfor 25 Marks	1. Water conservation techniques–Role of differents take holders–Implementation. 2. Preparing an action able business proposal for enhancing the village in come and approach for implementation. 3. Helping local schools to achieve good results and enhance their enrolment inHigher/technical/ vocational education.
5thSemfor 25 Marks	1. Developing Sustainable Water management systems for rural areas and implementation approaches. 2. Contribution to any national-level initiative of the Government of India. For eg. Digital India, Skill 3. India, SwachhBharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc. 4. Spreading public awareness under rural out reach programs. (minimum5programs). 5. Social connectand responsibilities.
6thSemfor 25 Marks	1. Plantation and adoption of plants.Know your plants. 2. Organize National integration and social harmony events/workshops/seminars. (Minimum02 programs). 3. Govt.schoolRejuvenationandhelpingthemtoachievegood infrastructure.

Course Outcomes (CourseSkillSet):

Attheend of the course, the studentwillbeableto:

- M23BNSK410.1Understandtheimportanceofhis/herresponsibilitiestowardssociety.
- M23BNSK410.2 alysetheenvironmentalandsocietalproblems/issuesandwillbeabletodesignsolutionsforthe same.
- M23BNSK410.3Evaluatetheexistingsystemand to propose practicalsolutions forthesame for sustainabledevelopment. M23BNSK410.4Implement government or self-driven projects effectively in the field.
- M23BNSK410.5Developcapacityto meet emergenciesandnatural disasters&practicenational integrationandsocial harmonyin general.

Pedagogy–Guidelines:

It may differ depending on local resources available for the study as well as environment and climatic differences, location, and time of execution.

Sl. No	Topic	Group size	Location	Activity execution	Reporting	Evaluation of the Topic
1.	Organicfarming, Indian Agriculture (Past, Present, and Future) Connectivityfor marketing.	May be individual or team	Farmers land/ Villages/ roadside/ communityarea/Collegecampus etc	Siteselection / Proper consultation/ Continuous monitoring/ Information board	Report should besubmittedby an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
2.	Waste management–Public, Private and Govt organization, 5R's.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes	Siteselection / proper consultation/ Continuous monitoring/ Information board	Report should be submitted by an individual to the concerned	Evaluation as per the rubrics of the scheme and syllabus

Dean Academics MIT Mysore

APPROVED Dean Academics MIT MYSORE

Principal MIT Mysore

			officers /campus etc…		evaluation authority	by NSSofficer
3.	Setting of the information imparting club for women leadingtocontribution in social andeconomicissues.	May be individual or team	Women empowerment groups/ Consulting NGOs&Govt Teams / Collegecampus etc…	Groupselection/ proper consultation / Continuous monitoring / Information board	Report should besubmittedby an individual to the concerned evaluation authority	Evaluation aspertherubricsof the scheme and syllabus by NSSofficer
4.	Water conservation techniques – Role ofdifferentstakeholdersImplementation.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers/campusetc…	Siteselection / Proper consultation/ Continuous monitoring/ Information board	Report should besubmittedby an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
5.	Preparinganactionable business proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers/campusetc…	Groupselection/ proper consultation / Continuous monitoring / Information board	Report should besubmittedby an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
6.	Helping local schools toachievegoodresults and enhance their enrolment in Higher/ technical/ vocational education.	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemesofficers /campusetc…	Schoolselection/ proper consultation / Continuous monitoring / Information board	Report should besubmittedby an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
7.	Developing Sustainable Water management system for rural areas and implementation approaches.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /campus etc…	Site selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
8.	Contribution to any national-level initiative of the Government of India. For eg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemes officers	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer

	development programs etc.		/ campus etc…			
9.	Spreading public awareness under rural outreach programs.(minimum 5 programs).Social connect and responsibilities.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
10.	Plantation and adoption of plants. Know your plants.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
11.	Organize National integration and social harmony events /workshops /seminars. (Minimum 02 programs).	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
12.	Govt. school Rejuvenation and helping them to achieve good infrastructure.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers /campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer

PlanofAction((ExecutionofActivitiesForEachSemester)

Sl. No	**PracticeSession Description**
1.	LecturesessionbyNSSOfficer
2.	Students'PresentationonTopics
3.	Presentation-1, Selection of topic, PHASE-1
4.	Commencementofactivityanditsprogress -PHASE-2
5.	Execution of Activity
6.	Execution of Activity
7.	Execution of Activity
8.	Execution of Activity
9.	Execution of Activity
10.	Casestudy-based Assessment, Individual performance
11.	Sectorwisestudyanditsconsolidation
12.	Video-basedseminarfor10minutesbyeachstudentAttheendofthesemesterwitha Report.

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

- In every semester from 3rd semester to 6th semester, Each student should do activities according to the scheme and syllabus.
- At the end of every semester student performance has to be evaluated by the NSS officer for theassigned activity progress and its completion.
- Atlastin6thsemesterconsolidatedreportofallactivitiesfrom3rdto6thsemester,compiled reportshouldbe submitted as per the instructions.

AssessmentDetails:

Weightage	**CIE- 100%**	
Presentation-1 Selection of topic, PHASE-1	**10 Marks**	• Implementation strategies of the project (NSS work). • ThelastReportshouldbesignedbytheNSS Officer, the HOD, and the principal. • AtlastReportshouldbeevaluatedbytheNSS officer of the institute. • Finally, the consolidated marks sheet should besenttotheuniversityandmadeavailableat the LIC visit.
Commencementofactivityanditsprogress - PHASE-2	**10 Marks**	
CaseStudy-basedAssessment	**10 Marks**	
Sector-wisestudy&its consolidation	**10 Marks**	
Videobasedseminarfor10minutesbyeach studentAttheendofsemesterwithReport. Activities.	**10 Marks**	
Totalmarksforthecourseineachsemester	**50 Marks**	

Marks scored for 50 by the students should be Scale down to 25 marks In each semester for CIE entry in the VTU portal.

25 marks CIE entry will be entered in University IA marks portal at the end of each semester 3rd to 6th sem, Report and assessment copy should be made available in the department semester wise

Students should present the progress of the activities as per the schedule in the prescribed practical session in thefield.
Thereshouldbepositiveprogressintheverticalorderforthebenefitofsocietyingeneral.

SuggestedLearningResources:
Books:

- NSSCourseManual,PublishedbyNSSCell,VTUBelagavi.
- GovernmentofKarnataka,NSScell,activitiesreportsandmanual.
- GovernmentofIndia,NSScell,Activitiesreportsandmanual.

4th Semester	**National Service Scheme /Physical Education/Yoga(NCMC)** **PHYSICAL EDUCATION (SPORTS & ATHLETICS) — II**	**M23BPEK410**

Semester-IV			
PHYSICAL EDUCATION (SPORTS& ATHLETICS) — I			
CourseCode	**M23BPEK410**	CIEMarks	**100**
NumberofLectureHours/Week(L: T:P:S)	**(0:0:2:0)**	SEEMarks	-
TotalNumberofLectureHours	-	TotalMarks	**100**
Credits	**0**	ExamHours	**-**

CourseOutcomes: At the end of the course, the student will be able to

- Understand the fundamental concepts and skills of Physical Education, Health, Nutrition and Fitness.
- Familiarization of health-related Exercises, Sports for overall growth and development.
- Create a foundation for the professionals in Physical Education and Sports.
- Participateinthecompetitionatregional/state/national/internationallevels.
- Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Module-1

Orientation **(5hours)**

- Lifestyle
- Fitness
- Food & Nutrition
- Health&Wellness
- Pre-Fitnesstest.

Module-2

GeneralFitness&ComponentsofFitness: **(15hours)**

- Warmingup(FreeHandexercises)
- Strength—Push-up/Pull-ups
- Speed—30MtrDash
- Agility—Shuttle Run
- Flexibility—Sitand Reach
- CardiovascularEndurance—HarvardstepTest

Module-3

RecreationalActivities: **(10hours)**

- Posturaldeformities.
- Stressmanagement.
- Aerobics.
- TraditionalGames.

Scheme and Assessment for auditing the course and Grades:

Sl. No.	**Activity**	**Marks**
1.	Participationofstudentinallthemodules	**20**
2.	Quizzes—2, each of 15marks	**30**
3.	Finalpresentation/exhibition/Participationincompetitions/practicalonspecifictasks assignedtothe students	**50**
	Total	**100**

Semester-IV

PHYSICAL EDUCATION(SPORTS&ATHLETICS)—II

Course Outcomes: At the end of the course, the student will be able to

M23BPEK410.1Understand the ethics and moral values in sports and athletics

M23BPEK410.2 Performintheselectedsportsorathleticsofthestudent'schoice.

M23BPEK410.3Understand the roles and responsibilities of organization and administration of sports and games.

Module-1

Ethics and Moral Values (5hours)

1. EthicsinSports
2. MoralValuesinSportsand Games

Module-2

Specific Games (Anyone to be selected by the student) (20hours)

1. Volleyball—Attack,Block,Service,UpperHandPassandLowerhandPass.
2. Throwball—Service,Receive,Spinattack,NetDrop&Jumpthrow.
3. Kabaddi—Handtouch,ToeTouch,ThighHold,AnkleholdandBonus.
4. Kho-Kho—GivingKho,SingleChain,Poledive,Poleturning,3-6Up.
5. TableTennis—Service(ForeHand&BackHand),Receive(ForeHand&BackHand),Smash.
6. Athletics(Track/FieldEvents)—AnyeventasperavailabilityofGround.

Module-3

Role of Organization and administration (5 hours)

SchemeandAssessmentforauditingthecourseand Grades:

Sl. No.	Activity	Marks
1.	Participation of student in all the modules	**20**
2.	Quizzes—2, each of 15marks	**30**
3.	Final presentation/exhibition/Participationincompetitions/practicalonspecifictasks assigned to the students	**50**
	Total	**100**

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

4th Semester	**National Service Scheme /Physical Education/Yoga (NCMC)** **Yoga**	**M23BYOK410**

Yoga			
Course Code	**M23BYOK410**		
NumberofLectureHours/Week(L:T:P:S)	**(0:0:2:0)**	CIEMarks	**100**
TotalNumberof LectureHours	-	SEEMarks	**-**
Credits	**0**	TotalMarks	**100**
Evaluation Method: Objective type Theory/Practical/Viva-oce			

Courseobjectives:

1. To enable the student to have good Health.
2. To practice mental hygiene.
3. To possesse motional stability.
4. To integrate moral values.
5. To attain a higher level of consciousness.

The Health Benefits of Yoga

The benefits of various yoga techniques have been supposed to improve

- bodyflexibility,
- performance,
- stressreduction,
- attainment of inner peace, and
- self-realization.

The system has been advocated as a complementary treatment to aid the healing of severalailmentssuchas

- coronaryheart disease,
- depression,
- anxietydisorders,
- asthma, and
- extensive rehabilitation for disorders including musculoskeletal problems and traumatic brain injury.

The system has also been suggested as behavioral therapy for smoking cessation and substance abuse (including alcohol abuse).

If you practice yoga, you may receive these physical, mental, and spiritual benefits:

- Physical
  1. Improved body flexibility and balance
  2. Improved cardiovascular endurance (strongerheart)
  3. Improved digestion
  4. Improved abdominal strength
  5. Enhanced overall muscular strength
  6. Relaxation of muscularstrains
  7. Weight control
  8. Increased energy levels
  9. Enhanced immune system
- Mental
  1. Relief of stress resulting from the control of emotions
  2. Prevention and relief from stress-related disorders
  3. Intellectual enhancement, leading to improved decision-making skills
- Spiritual
  1. Life withm eaning, purpose, and direction
  2. Inner peace and tranquility
  3. Contentment

YogaSyllabus

SemesterIII

- Yoga, its origin, history and development. Yoga, its meaning, definitions.
- Different schools of yoga, Aim and Objectives of yoga, importance of prayer
- Yogic practices for a common man to promote positive Health
- Rules to be followed during yogic practices by the practitioner
- Yoga its misconceptions,
- Difference between yogic and non-yogic practices
- Suryanamaskar prayer and its meaning, Need, importance and benefits of Suryanamaskar 12 count, 2rounds
- Asana, Need, importance of Asana. Different types of asanas. Asana its meaning by name, technique, precautionary measures and benefits of each asana
- DifferenttypesofAsanas
  - a. Sitting
    1. Padmasana
    2. Vajrasana
  - b. Standing
    1. Vrikshana
    2. Trikonasana
  - c. Proneline
    1. Bhujangasana
    2. Shalabhasana
  - d. Supineline
    1. Utthitadvipadasana
    2. Ardhahalasana

Semester IV

- Patanjali's Ashtanga Yoga, its need and importance.
- Yama: Ahimsa, satya, asteya, brahmacarya, aparigraha.
- Niyama: shoucha, santosh, tapa, svaadhyaya, Eshvarapranidhan
- Suryanamaskar12count-4roundsofpractice
- Asana, Need, importance of Asana. Different types of asana. Asana its meaning by name, technique, precautionary measures and benefits of each asana.
- Different types of Asanas
  - a. Sitting
    1. Sukhasana
    2. Paschimottanasana
  - b. Standing
    1. Ardhakati Chakrasana
    2. Parshva Chakrasana
  - c. Prone line
    1. Dhanurasana
  - d. Supine line
    1. Halasana
    2. Karna Peedasana
- Meaning, importance and benefits of Kapalabhati.
- 40 strokes/min 3 rounds
- Meaning, Need, importance of Pranayama. Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama.
- Pranayama
  1. Suryanuloma –Viloma
  2. Chandranuloma-Viloma
  3. Suryabhedana
  4. Chandra Bhedana
  5. Nadishodhana

Semester V

- Patanjali'sAshtanga Yoga its need and importance.
- Ashtanga Yoga
  1. Asana
  2. Pranayama
  3. Pratyahara
- Asana its meaning by name, technique, precautionary measures and benefits of each asana
- Different types of Asanas
  - **a.** Sitting 1. Ardha Ushtrasana 2. Vakrasana 3. Yogamudra in Padmasana
  - **b.** Standing 1. UrdhvaHastothanasana 2. Hastapadasana 3. ParivrittaTrikonasana 4. Utkatasana
  - **c.** Prone line 1. Padangushtha Dhanurasana 2. Poorna Bhujangasana / Rajakapotasana
  - **d.** Supine line 1. Sarvangasana 2. Chakraasana 3. Navasana/Noukasana 4. Pavanamuktasana
- Revision of practice 60 strokes/min 3 rounds
- Meaning by name, technique, precautionary measures and benefits of each Pranayama 1. Ujjayi 2. Sheetali3. Sheektari

Semester VI

- **Ashtanga Yoga**
  1. **Dharana**
  2. Dhyana (Meditation)
  3. Samadhi
- Asana by name, technique, precautionarymeasuresandbenefitsofeach asana
- DifferenttypesofAsanas
  - a. Sitting1.Bakasana 2. Hanumanasana 3. Ekapada Rajakapotasana 4. YogamudrainVajrasana
  - b. Standing1. Vatayanasana2. Garudasana
  - c. Balancing1.Veerabhadrasana2. Sheershasana
  - d. Supineline1.Sarvangasana 2. Setubandha Sarvangasana 3. Shavasanaa (Relaxationpoisture).
- Revision of Kapalabhati practice80strokes/min-3rounds
- Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama 1. Bhastrika 2. Bhramari
- Meaning, Need, importance of Shatkriya.
  - **2.** Different types. Meaning by name, technique, precautionary measures and benefits of each Kriya 1. Jalaneti &sutraneti 2. Nouli (onlyfor men) 3. SheetkarmaKapalabhati

Course outcomes (CourseSkillSet):
At the end of the course, the student will be able to:

- **M23BYOK410.1:** Underst and the meaning, aim and objectives of Yoga.
- **M23BYOK410.2:** Perform Suryanamaskar and able toTeach its benefits.
- **M23BYOK410.3:** Understand and teach different Asanas by name, its importance, methods and benefits.
- **M23BYOK410.4:** Instruct Kapalabhati and its need and importance.
- **M23BYOK410.5:** Teach different types of Pranayama by its name, precautions, procedure and uses
- **M23BYOK410.6:** Coach different types of Kriyas, method to follow and usefulness.

Assessment Details (both CIE and SEE)

- Students will be assessed with internal test by a. Multiple choice questions b. Descriptive type questions (Two internal assessmenttests with 25 marks/test)
- Final test shall be conducted for whole syllabus for 50marks.
- ContinuousInternal Evaluationshallbefor100marks(includingIAtest)

SuggestedLearningResources:
Books:

1. YogapraveshainKannadabyAjitkumar
2. Light onYogabyBKS Iyengar
3. Teaching Methods for Yogic-practices by Dr.MLGharote&Dr.SK Ganguly
4. Yoga Instructor Course handbook published by SVYASA University, Bengaluru
5. YogaforChildren–stepbystep –byYamini Muthanna

Web links and Video Lectures
(eResources):Referlinks

1. https://youtu.be/KB-TYlgd1wE
2. https://youtu.be/aa-TG0Wg1Ls

4th Semester	**Basic Science Course (BS)** **DIPLOMA MATHEMATICS-II**	**M23BDIPM411**

1. Prerequisites

S/L	Proficiency	Prerequi sites
1	Linear Algebra	Linear algebra is a foundational subject in mathematics with wide-ranging applications in science, engineering, computer science, economics, and more. To effectively learn and understand linear algebra, it is important to have Basic Algebra, Geometry, familiarity with summation notation, matrix notation, and other mathematical symbols used in linear algebra.
2	Higher-Order Differential Equations	To effectively learn and understand higher-order differential equations, one should have a solid foundation in several mathematical areas like Algebraic Manipulations, Differentiation, First-Order ODEs, Familiarity with solving second-order linear differential equations with constant coefficients, including homogeneous and non-homogeneous cases and Partial Fraction.
3	Probability Theory Higher-Order Differential Equations	Probability theory is an essential subject for engineering students, as it provides the foundation for understanding and modeling uncertainty in various engineering applications. Comfort with simplifying and solving algebraic equations, Understanding of basic counting principles, such as the multiplication rule, permutations, and combinations.
4	Numerical Methods-I & II	Strong foundation in calculus, linear algebra, and basic programming skills. Understanding concepts such as differentiation, integration, matrices, vectors, and algorithms is essential for effectively applying numerical methods in solving mathematical problems.
5	Previous Coursework	Completion of introductory courses in Mathematics or a related field.

2. Competencies

S/L	Competency	KSA Description
1	Linear Algebra	**Knowledge:**Understand the concept of an inverse matrix and how to find it (if it exists), solving systems of linear equations, such as Gaussian elimination and matrix inversion, Understand row reduction techniques and the concepts of row echelon form (REF) and reduced row echelon form (RREF). **Skills:** Studying linear algebra effectively requires a combination of specific skills and Analytical Skills to develop the ability to approach and solve a variety of linear algebra problems systematically, **Attitude:** Understanding its practical utility can make the subject more engaging and relevant.
2	Higher-Order Differential Equations	**Knowledge:** Understand what constitutes a higher-order differential equation and the significance of the order, differences and implications of homogeneous and non-homogeneous equations, learn to form and solve the characteristic equation to find the general solution of homogeneous equations, **Skills:** Develop the ability to systematically approach and solve a variety of differential equations, Recognize and understand the applications of differential equations in other areas of engineering, such as mechanical vibrations, electrical circuits, and control systems, develop skills to model real-world engineering problems using differential equations. **Attitude:** It can significantly enhance your learning experience and success in studying higher-order differential equations, some of them are
3	Probability Theory	**Knowledge:** Understanding of basic probability concepts including sample spaces, events, and the axioms of probability, Familiarity with probability rules such as addition and multiplication rules. **Skills:** Develop systematic approaches to solving probability problems, Practice breaking down complex problems into simpler parts. Enhance the ability to critically evaluate probabilistic models and assumptions. **Attitude:** Develop an interest in how probability theory applies to real-world engineering problems, such as reliability analysis, quality control, and risk

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

		assessment.
4	Numerical Methods-I & II	**Knowledge:** Students will learn various numerical approximation techniques, such as interpolation, curve fitting, and numerical differentiation and integration, which are essential for approximating functions and data in engineering analysis. **Skills:**It helps to acquire practical skills and knowledge that are essential for solving complex engineering problems that may not have analytical solutions. Solving complex engineering problems, analyzing mechanical systems, and optimizing design processes using computational tools and simulations. **Attitude:** Methodical approach to testing and validating numerical algorithms for accuracy and efficiency. Adaptability to new tools, libraries, and frameworks that facilitate numerical computations.

3. Syllabus

Diploma Mathematics-II
Semester-IV

Course Code	**M23BDIPM411**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:0:0:0)**	SEE Marks	**00**
Total Number of Lecture Hours	**20 Hours**	Total Marks	**50**
Credits	**00**	Exam Hours	**00**

Course objectives:
The mandatory course M23BDIPM411viz., Diploma Mathematics –IIaims to provide essential concepts of Linear algebra, Second and higher-order differential equations, insight into Elementary probability theory and Numerical methods.

Module -1: Linear Algebra

Introduction, Rank of a matrix by elementary row operations, Consistency of system of linear equations, Solution by Gauss Elimination method. Eigenvalues and eigenvectors of a square matrix. Problems.

Module -2: Higher-Order Differential Equations

Linear homogeneous/nonhomogeneous differential equations of second and higher-order with constant coefficients. Solution by using the inverse differential operator method.

Module -3: Probability Theory

Introduction, Sample space and Events, Axioms of Probability. Addition and Multiplication theorem. Conditional Probability. Independent events. Baye's theorem, Problems.

Module -4: Numerical Methods -1

Finite differences, Interpolation/extrapolation using Newton's forward and Backward difference formulae (No derivation), Problems. Solution of polynomial and transcendental equations by Newton–Raphson and Regula–Falsi methods (no derivation), Problems. Numerical Integration: Simson's 1/3 rd rule and 3/8 rule, problems.

Module -5: Numerical Methods -2

Numerical solution of first-order ordinary differential equations: Taylor's series method, Modified Euler's method, Runge-Kutta method of order 4, Milne's predictor-corrector method. Problems.

Text Books:

1. Higher Engineering Mathematics: B. S. Grewal, Khanna Publishers, New Delhi, 43rd Ed., 2015.
2. Higher Engineering Mathematics: V. Ramana, McGraw-Hill Education, 11th Ed.

Reference Books:

1. Engineering Mathematics: Srimanta Pal & Subodh C. Bhunia, Oxford University Press, 3rd Reprint, 2016.
2. A textbook of Engineering Mathematics: N.P Bali and Manish Goyal, Laxmi Publications, Latest edition.
3. Higher Engineering Mathematics: H.K. Dass and Er. Rajnish Verma, S. Chand Publication (2014).

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Linear Algebra	Introduction Rank of a matrix by elementary row operations Consistency of system of linear equations Problems Solution by Gauss Elimination method. problems Eigenvalues and eigenvectors of a square matrix.

		Problems.
2	Week 3-4: Higher-Order Differential Equations	Linear homogeneous Complementary function Problems Non-homogeneous differential equations Solution by using the inverse differential operator method. Particular method for e^{ax} Particular method for sinax/ cosax Particular method for x^{n}
3	Week 5-6: Probability Theory	Introduction, Sample space and Events, Axioms of Probability. Addition and Multiplication theorem. Conditional Probability. Independent events. Baye's theorem, Problems.
4	Week 7-8: Numerical Methods -1	Solution of algebraic and transcendental equations - Regula-Falsi and Newton-Raphson methods, Problems. Finite differences, Interpolation using Newton's forward and backward difference formulae. Newton's divided difference formula Lagrange's interpolation formula. Problems. Numerical integration: Trapezoidal, Simpson's $(1/3)^{rd}$ and $(3/8)^{th}$ rules. Problems.
5	Week 9-10: Numerical Methods -2	Numerical solution of ordinary differential equations of first order and first degree - Taylor's series method. Problems Modified Euler's method Problems. Runge-Kutta method of fourth order. Problems. Milne's predictor-corrector formula. Problems.
6	Week 11-12: Integration and Practical Applications	Apply learned concepts and competencies to real-world scenarios. Hands-onpractice

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 100%. The minimum passing mark for theCIE is 40% of the maximum marks (100). A student shall be deemed to have satisfied the academic. Requirements if the student secures not less than 40% (40 Marks out of 100) in the CIE.

Continuous Internal Evaluation:

Components		Number	Weightage	Max. Marks	Min. Marks
(i)	Internal Assessment-Tests (A)	2	50%	25	10
(ii)	Assignments/Quiz/Activity (B)	2	50%	25	10
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

Average internal assessment shall be the average of the 2 test marks conducted.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Linear Algebra	Linear algebra is used to model and analyze dynamic systems, such as electrical circuits, mechanical systems, and chemical processes. Techniques like matrix operations, eigenvalues, and eigenvectors help engineers understand system behavior and design controllers for optimal performance.
2	Higher-Order Differential Equations	Engineers use higher-order differential equations to model the motion of mechanical systems such as vibrating structures, rotating machinery, and vehicles, also used to describe the behavior of electrical circuits, including the flow of current and voltage across different components.
3	Probability Theory	Probability theory in engineering is a mathematical framework used to model and analyze uncertainty in engineering systems. It provides tools for quantifying the likelihood of various outcomes and understanding the behavior of complex systems under uncertain conditions.
4	Numerical Methods	Numerical integration methods, such as the trapezoidal rule, Simpson's rule are used to approximate definite integrals. Numerical differentiation methods, such as finite differences, are used to estimate derivatives.

8. Course Outcomes (COs) and Mapping with POs/ PSOs

Course Outcomes (COs)

COs	Description
M23BDIPM411.1	Apply elementary probability theory; solve related problems on consistency and system of linear equations.
M23BDIPM411.2	Apply numerical methods in modeling and the concept of higher order differential equations for solving engineering problems.
M23BDIPM411.3	Analyze the Engineering application problem through Numerical technique.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
M23BDIPM411.1	**3**	**-**									**3**	
M23BDIPM411.2	**3**										**3**	
M23BDIPM411.3		**3**										**3**
M23BDIPM411	**3**	**3**									**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	**Total**
Module 1	**2**	**3**	**5**	**10**
Module 2	**2**	**3**	**5**	**10**
Module 3	**2**	**3**	**5**	**10**
Module 4	**2**	**3**	**5**	**10**
Module 5	**2**	**3**	**5**	**10**
Total	**10**	**15**	**25**	**50**

10. Future with this Subject

The “Diploma Mathematics-II "course in the second year of the B.E program has strong foundation for several future courses in the undergraduate program. The future for engineering students who study subjects like linear algebra, higher-order differential equations, probability theory, and numerical methods is promising and filled with opportunities. Here's why:

Industry Demand:

Industries across various sectors, including aerospace, automotive, electronics, and energy, rely heavily on mathematical modeling and analysis. Proficiency in subjects like linear algebra, differential equations, probability theory, and numerical methods is essential for solving complex engineering problems in these industries.

Advanced Technology and Innovation:

With the rapid advancement of technology, engineering solutions are becoming increasingly complex. Skills in mathematical modeling and computational techniques are crucial for developing innovative technologies and solutions. Knowledge of these mathematical subjects is not limited to a single engineering discipline but finds applications across various fields.Engineering students with a strong foundation in these subjects can explore interdisciplinary opportunities and collaborate on projects that require diverse skill sets.

Research and Development:

In research and development (R&D) roles, engineers often encounter complex mathematical problems that require advanced analytical and computational techniques. Proficiency in subjects like linear algebra, differential equations, and numerical methods is essential for conducting impactful research and developing innovative solutions.