MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Autonomous Institution Affiliated to VTU

Competency-Based Syllabus Design

For

Information Science and Engineering (IS&E)

(Under Outcome Based Education (OBE) and

Choice-Based Credit System (CBCS))

Offered from 5th to 6th Semester of Study

In

Partial Fulfillment for the Award of Bachelor's Degree

Information Science and Engineering

2023 scheme

Scheme Effective from the academic year 2023-24

Contents

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

5th Semester	**Professional Core Course (PC)** **Theory of Computation**	**M23BCS501**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Mathematical Foundations	Discrete Mathematics: Understanding topics such as sets, relations, functions, combinatorics, logic, and proof techniques (e.g., induction, contradiction) is crucial. Graph Theory: Basic knowledge of graphs, trees, and their properties. Linear Algebra: While not always essential, it can be helpful in certain areas like quantum computation. Number Theory: Basics of primes, divisibility, and modular arithmetic can be useful.
2	Formal Logic	Propositional Logic: Basic logical operations, truth tables, and logical equivalences. Predicate Logic: Understanding of quantifiers, predicates, and logical inference.
3	Programming and Algorithms	Data Structures: Familiarity with common data structures like arrays, lists, stacks, queues, trees, and graphs. Algorithms: Basic knowledge of algorithm design, analysis (time and space complexity), and common algorithms (e.g., sorting, searching). Programming: Proficiency in at least one programming language is helpful for implementing and understanding theoretical concepts.
4	Automata and Formal Languages	Finite Automata: Basics of deterministic and nondeterministic finite automata. Regular Languages: Understanding of regular expressions and their relation to finite automata. Context-Free Grammars: Basics of context-free grammars and pushdown automata.
5	Theoretical Understanding	Grasp abstract concepts and their implications for computation
6	Problem-Solving Skills	Apply theoretical concepts to solve computational problems.
7	Basic Understanding of Computability and Complexity	Turing Machines: Introduction to the concept of Turing machines and the Church-Turing thesis. P vs NP Problem: A basic understanding of computational complexity classes like P, NP, and NP-complete problems.

2. Competencies

S/L	Competency	KSA Description
1	**Formal Language Theory**	**Knowledge:** It includes understanding fundamental concepts in formal languages, grammars, automata, and the relationships between them. It also encompasses the theoretical foundation needed to understand how formal languages are defined and processed. **Skills:** It involves the ability to apply theoretical knowledge to solve problems and work with formal models. This includes working with formal grammars, automata, and performing operations on languages to solve computational problems.. **Attitude:** It refers to the mindset or approach an individual adopts when engaging with formal language theory. These attitudes foster deep understanding, critical thinking, and a responsible application of theoretical knowledge.
2	**Automata Theory**	**Knowledge:** Understand the fundamental concepts, theories, and frameworks that underlie automata and formal languages. **Skills:** Ability to apply the knowledge of automata theory to real-world problems, design automata, and perform formal proofs.

		Attitude: It refers to the mindset and approach necessary to approach problems in automata theory with rigor, creativity, and persistence.
3	**Computability**	**Knowledge:** Competency in computability requires foundational knowledge in several areas of computer science and mathematics. **Skills:** In addition to theoretical knowledge, a competent person in computability must develop practical skills such as Problem-Solving with Formal Models, Algorithm Design and Proof, Mathematical Proof Techniques etc. **Attitude:** It is having the right attitudes is key for competency in computability such as critical thinking, curiosity, precision and rigor.
4	**Complexity Theory**	**Knowledge:** To be competent in complexity theory, a deep understanding of foundational concepts is necessary. **Skills:** Being competent in complexity theory not only requires knowledge but also the ability to apply that knowledge to solve real problems. **Attitude:** Having the right mindset and attitudes is key to achieving competency in complexity theory.
5	**Parsing and Language Processing**	**Knowledge:** Requires an understanding of both **formal language theory** and **computational techniques** used for analyzing and generating language. **Skills:** To be proficient in parsing and language processing, you need to be able to apply theoretical knowledge to practical problems. **Attitude:** The right attitudes are crucial for working with complex language processing tasks
6	**Formal Verification**	**Knowledge:** Using mathematical techniques we have to prove that a system satisfies certain properties, such as correctness, security, or safety, under all possible conditions. This requires a deep understanding of formal methods, logic, and verification techniques, along with the ability to apply these techniques in practice. **Skills:** To effectively apply formal verification techniques, we need both theoretical and practical skills. **Attitude:** The right mindset is essential for successfully applying formal verification, as it involves complex, sometimes difficult, problem-solving.

3. Syllabus

THEORY OF COMPUTATION **SEMESTER – V**			
Course Code	**M23BCS501**	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:

1. Introduce core concepts in Automata and Theory of Computation
2. Identify different Formal language Classes and their Relationships
3. Design Grammars and Recognizers for different formal languages
4. Prove or disprove theorems in automata theory using their properties
5. Determine the decidability and intractability of Computational problems

Module -1

Introduction to Finite Automata, Structural Representations, Automata and Complexity. The Central Concepts of Automata Theory. Deterministic Finite Automata, Nondeterministic Finite Automata, An Application: Text Search, Finite Automata with Epsilon-Transitions.
Textbook 1:**Ch 1.1,1.5,2.2,2.3,2.4,2.5**

Module -2

Regular Expressions, Finite Automata and Regular Expressions, Proving Languages not to be Regular. Closure Properties of Regular Languages, Equivalence and Minimization of Automata, Applications of Regular Expressions
Textbook 1:**Ch** 3.1, 3.2 (Except 3.2.1), 3.3, 4.1, 4.2, 4.4

Module -3

Context-Free Language : Context-Free Language and Derivation trees, Ambiguity in context-free grammars, simplifying of context-free grammars, Normal forms for CFGs
Textbook 2:**Ch 6.1-6.4,7.1**

Module -4

Pushdown Automata (PDA): Definition of the Pushdown Automaton, The Languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.
Properties of Context-Free Languages: The pumping lemma for CFGs; Closure properties of CFLs
Textbook 1:**Ch 6.1,6.2,6.3.1,6.4,7.2,7.3**

Module -5

Turing Machine: Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. Variants of Turing Machines (TM), The model of Linear Bounded automata
Decidability: Definition of decidability, decidable languages, Undecidable languages, halting problem of TM, Post correspondence problem.
Textbook 2:**Ch 9.1-9.8,10.1-10.7**

TEXTBOOKS:

1. John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman," Introduction to Automata Theory, Languages and Computation", Second Edition, Pearson.
2. K L P Mishra, N Chandrasekaran , 3rd Edition, Theory of Computer Science, PhI, 2012.Marketing Management: A relationship approach (2019), Hollensen, S, Pearson Education.

REFERENCE BOOKS:

1. Elain Rich, "Automata, Computability and complexity", 1st Edition, Pearson Education, 2018.
2. Michael Sipser : Introduction to the Theory of Computation, 3rd edition, Cengage learning,2013
3. John C Martin, Introduction to Languages and The Theory of Computation, 3rd Edition, Tata McGraw – Hill Publishing Company Limited, 2013
4. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012

Faculty can utilize open source tools (like JFLAP) to make teaching and learning more interactive.

4. Syllabus Timeline

S/L	**Syllabus Timeline**	**Description**
1	Week 1 Introduction to Theory of Computation	Overview of computation models, Historical context and significance and Introduction to formal languages and automata
2	Week 2-4: Finite State Machines	Understand the concepts and how to design Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA),ε-NFA and Minimizing FSMs, construct Finite State Transducers and Bidirectional Transducers.
3	Week 5-6: Regular Expressions Regular Grammars	Understand the concepts of Regular expressions and Regular grammar and their equivalence with finite automata, Properties of Regular Languages.
4	Week 7-8: Context-Free Grammars	Understand the concept of Context-Free Grammars (CFG),Parse trees and derivations, Normal Form and how to eliminate ambiguity in grammar.

5	Week 9-10: Pushdown Automata	Definition of the Pushdown automata, understand the languages of a PDA; conversion of PDA's and CFG's; Prove that given PDA is Deterministic or not. Understand the Properties of Context-Free Grammar
6	Week 11: Turing machines	Introduction to Turing machines, Representation TM, designing of TM Variants of Turing machines (multi-tape, nondeterministic, etc.)
7	Week 12: Decidability	Understand the concept of decidability and undecidable, The Halting problem, Post correspondence problem.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Start with Real-World Examples	**Teaching Approach:** Begin by introducing concepts using real-world analogies or simple, relatable examples. For instance, explain finite automata using examples like traffic light systems or vending machines. **Learning Approach:** Encourage students to think of everyday processes or systems that can be modeled by the concepts being studied.
2	Problem-Based Learning (PBL)	**Teaching Approach:** Present students with problems that require them to apply theoretical concepts to find solutions. This could involve designing automata for specific tasks, proving language properties, or reducing problems to show complexity classifications. **Learning Approach:** Engage in active problem-solving during and outside class. Form study groups to tackle challenging problems collectively, enhancing understanding through discussion and collaboration.
3	Use of Visual Aids and Diagrams	**Teaching Approach:** Incorporate diagrams, flowcharts, and other visual aids to explain abstract concepts. For instance, use state diagrams to represent finite automata or Turing machines. **Learning Approach:** Create your own visual representations of concepts as a study tool. Drawing out problems and solutions can help clarify complex ideas..
4	Collaborative Projects	**Teaching Approach:** Assign group projects where students can work together to explore a theoretical concept in depth or implement a computational model. **Learning Approach:** Collaborate effectively with peers, dividing tasks based on individual strengths while ensuring everyone understands the overall project.
5	Multimodal Teaching	**Teaching Approach:** Incorporate a mix of lectures, visual aids, interactive simulations, group work, and hands-on projects to cater to different learning styles. **Learning Approach:** Identify your preferred learning style and seek out resources or study methods that align with it, whether it's visual, auditory, reading/writing, or kinesthetic.

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

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
	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

Semester End Examination:

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 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 Formal Languages	Explain the concept of formal languages and distinguish between different classes of languages, such as regular, context-free, and context-sensitive languages.
2	Mastering Automata Theory	Design and analyze deterministic and nondeterministic finite automata (DFA and NFA) for recognizing regular languages. Understand and apply the concepts of context-free grammars (CFG) and pushdown automata (PDA) to recognize context-free languages.
3	Exploring Computability Theory	Understand the concept of Turing machines and their role in defining computation. Analyze the concepts of decidability and undecidability, including understanding the Halting problem and its implications.
4	Grasping Computational Complexity	Understand and differentiate between complexity classes such as P, NP, NP-complete, and NP-hard.
5	Applying Theoretical Concepts to Problem-Solving	Use formal methods to prove language properties, such as closure properties and the pumping lemma for regular and context-free languages and also develop problem-solving skills by applying theoretical concepts to real-world scenarios, such as designing algorithms based on automata or analyzing the computational complexity of tasks

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

Course Outcomes (COs)

Cos	Description
M23BCS501.1	Apply the fundamentals of automata theory to write DFA, NFA, Epsilon-NFA and conversion between them.
M23BCS501.2	Design context-free grammars (CFGs) and pushdown automata (PDAs) for formal languages
M23BCS501.3	Design Turing machines to solve the computational problems and also Illustrate the concepts of decidability and undecidability.
M23BCS501.4	Prove the properties of regular languages using regular expressions

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS501.1	-	3	-	-	-	-	-	-	-	-	-	2	3	-
M23BCS501.2	-	-	3	-	-	-	-	-	-	-	-	2	-	3
M23BCS501.3	-	-	3	-	-	-	-	-	-	-	-	2	-	3
M23BCS501.4	-	-	-	2	-	-	-	-	-	-	-	2	-	2
M23BCS501	**-**	**3**	**3**	**2**								**2**	**3**	**3**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**10**				**10**
Module 2				**10**	**10**
Module 3		**5**			**10**
Module 4		**5**			**10**
Module 5			**10**		**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		**10**			**20**
Module 4		**10**			**20**
Module 5			**20**		**20**
Total	**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

1. Advancements in Artificial Intelligence and Machine Learning

- Natural Language Processing (NLP): Automata theory is crucial in the development of NLP algorithms. Understanding regular expressions and finite automata helps in text parsing and tokenization, which are fundamental in building efficient NLP models.
- Pattern Recognition: Automata and formal languages are used in recognizing patterns in data, which is vital in developing AI systems that can learn and adapt.

2. Quantum Computing

- New Computational Models: As quantum computing advances, new models of computation are emerging. Automata theory provides the foundational knowledge necessary to understand and develop quantum automata, which could revolutionize how we process information.

3. Cybersecurity

- Formal Verification: Automata theory plays a significant role in formal methods, which are used to verify the correctness of security protocols and systems. As cybersecurity threats grow, the need for rigorous verification methods will increase.
- Intrusion Detection Systems: Automata-based models are used in developing algorithms for detecting and responding to security breaches, helping to enhance the security of systems.

4. Compiler Design and Programming Languages

- Compiler Optimization: Automata theory is integral to the design and optimization of compilers, which translate high-level programming languages into machine code. Future advances in programming languages will continue to rely on automata theory to improve compiler efficiency.
- Language Development: As new programming languages are developed, understanding automata and formal grammars will be key in designing languages that are both powerful and efficient.

5. Internet of Things (IoT) and Embedded Systems

- Finite State Machines (FSMs)**:** Automata theory underpins the design of finite state machines, which are crucial in the development of IoT devices and embedded systems. As these technologies become more widespread, the need for expertise in automata will grow.

5th Semester	**Integrated Professional Core Course(IPC)** **Computer Networks and Security**	**M23BCS502**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Computer Science Knowledge	Basic Computer Science Knowledge: Programming Skills: Familiarity with programming languages such as Python, C, or Java is essential for scripting and automation tasks. Operating Systems: Understanding of operating system concepts, including process management, memory management, and file systems. Data Structures and Algorithms: Knowledge of basic data structures (e.g., arrays, lists, trees) and algorithms (e.g., sorting, searching) is important for problem-solving and optimization.
2	Networking Fundamentals	OSI and TCP/IP Models: Understanding the OSI model and the TCP/IP stack, including the functions of different layers (physical, data link, network, transport, and application). IP Addressing and Sub-netting: Knowledge of IP address formats (IPv4 and IPv6), subnet masks, and how to perform sub-netting and address allocation. Network Protocols: Familiarity with key protocols such as HTTP, FTP, TCP, UDP, DNS, and DHCP. Routing and Switching: Understanding of how routers and switches operate, including basic routing algorithms and protocols like RIP, OSPF, and BGP.
3	Network Devices and Technologies	Network Devices: Knowledge of different network devices such as routers, switches, firewalls, and access points. Wireless Technologies: Understanding of wireless networking principles, including Wi-Fi standards and security measures.
4	Cyber-security Basics	Security Principles: Awareness of core security principles, including confidentiality, integrity, and availability. Common Threats and Attacks: Knowledge of common types of cyber -attacks, such as phishing, malware, ransom-ware, and denial-of-service (DoS) attacks. Cryptography: Basic understanding of cryptographic concepts like encryption, decryption, hashing, and digital signatures.
5	Network Security Fundamentals	Firewalls and Intrusion Detection Systems (IDS): Familiarity with how firewalls and IDS/IPS (Intrusion Prevention Systems) function and are configured. VPNs and Encryption: Understanding of Virtual Private Networks (VPNs) and encryption protocols used to secure network communications. Access Control: Knowledge of access control methods, including authentication, authorization, and accounting (AAA).
6	Hands-On Experience	Lab Practice: Practical experience with network configuration, management, and troubleshooting using tools like Cisco Packet Tracer or GNS3. Security Tools: Familiarity with security tools and techniques, such as network scanners (N-map), penetration testing tools (Metasploit), and security information and event management (SIEM) systems.

2. Competencies

S/L	Competency	KSA Description
1	**Application protocols**	**Knowledge:** Understand the basic functions of application protocols **Skills**: know the service provided by the DNS,HTTP,FTP **Attitudes:** Appreciation for the versatility of application protocols and distributed services.
2	**Transport layer**	**Knowledge:** Data volume, destination, and rate are all controlled by transport-layer protocols including TCP, UDP, DCCP, and SCTP. **Skills:** the transport layer receives the packets, sorts them, and looks for faults. Subsequently, Enable efficient network transmission,

		Attitudes: provide the communication services directly to the application processes running on different hosts
3	**Network layers**	**Knowledge:** The main job of this layer is to maintain the quality of the data and pass and transmit it from its source to its destination. There are several important protocols that work in this layer. **Skills:** Selects the best path to transfer the data from source to its destination. **Attitudes:** The network layer performs packetization on the data. This makes it easier to transmit the data packets in the network.
4	**Security services and attacks**	**Knowledge:** Mechanisms used to provide confidentiality, identity authentication, integrity authentication, source authentication, and/or support the non-repudiation of information. **Skills:** A mechanism that is designed to detect, prevent, or recover from a security attack. Security Service: A service that enhances the security of data processing systems and information transfers **Attitudes:** The versatility of security guards across various sectors—from retail to construction, events, and VIP protection, offering specialized services that go beyond general surveillance.
5	**Cryptography**	**Knowledge:** There are two main types of cryptography used for digital data and secure messages today: symmetric cryptography and asymmetric cryptography. Hash functions, a third type, doesn't involve use of a key **Skills:** Basically to protect their privacy and keep their conversations and data confidential. Cryptography ensures confidentiality by encrypting sent messages using an algorithm with a key only known to the sender and recipient. **Attitudes:** By employing cryptographic techniques, data security applications help to protect various everyday transactions, ranging from file sharing within business networks to online apps used for banking or shopping..
6	**Cloud computing and its security**	**Knowledge:** Associated with cloud computing and determine the best security measures for protecting data with cloud security models. **Skills:** Identify sensitive or regulated data. Understand how sensitive data is being accessed and shared. Discover shadow IT (unknown cloud use). **Attitudes:** Cloud computing gives your business more flexibility. You can quickly scale resources and storage up to meet business demands without having to invest in physical infrastructure.
7	**Transport level security**	**Knowledge:** Internet Engineering Task Force (IETF) standard protocol that provides authentication, privacy and data integrity between two communicating computer applications. **Skills:** Transport Security Layer (TLS) is the successor of the Secure Socket Layer (SSL); both are security protocols and are sometimes used interchangeably. **Attitudes:** TLS is usually implemented on top of **TCP** (Transmission Control Protocol) which further encrypts Application Layer protocols such as FTP, SMTP, and HTTP. However, it can be implemented on DCCP, UDP, and SCTP
8	**Web security**	**Knowledge:** Provides protection for web applications against attacks, including cross-site scripting, file inclusion, cross-site forgery, Structured Query Language (SQL) injection, and other threats. **Skills:** A web application firewall (WAF) protects web applications by monitoring and filtering internet traffic that flows between an application and the internet. In this way, a WAF works as a secure web gateway (SWG). **Attitudes:** Web scanning involves using an application to crawl a website in search for vulnerabilities that can leave it open to a bot, spyware, rootkit, Trojan horse, or distributed denial-of-service (DDoS) attack, It then systematically checks the entire site for potential weaknesses.
9	**IP security**	**Knowledge:** unique identifying number assigned to every device connected to the internet.

		Skills: To protect network data by setting up circuits using IPsec tunnelling in which all data being sent between the two endpoints is encrypted, as with a Virtual Private Network (VPN) connection. **Attitudes:** IP Security is a versatile and widely adopted security protocol used in various network scenarios, including site-to-site VPNs, remote access VPNs, and secure communication between network devices.

3. Syllabus

Computer Networks and Security
SEMESTER – V

Course Code	**M23BCS502**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:1:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory +20 hours Practical**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course Objectives:

1. To understand network application architectures and the many types of service protocols such as HTTP, SMTP, FTP, and DNS.
2. Analyze how the transport layer responds to network and application requests with various services and functions such as TCP and UDP.
3. To analyze the network layer and how it is related to routing and security between two data sets and its protocols, broadcast and multicast path.
4. Analyze security services and attacks by using data encryption and decryption techniques with the use of symmetric and asymmetric algorithms.
5. Apply and analyze the access control and network security like cloud security. Transport layer security, web security and IP security.

Module -1

Application Layer: Principles of Network Applications: Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols. The Web and HTTP: Overview of HTTP, Non-persistent and Persistent Connections, HTTP Message Format, User-Server Interaction: Cookies, Web Caching, The Conditional GET, File Transfer: FTP Commands & Replies, Electronic Mail in the Internet: SMTP, Comparison with HTTP, Mail Message Format, Mail Access Protocols, DNS; The Internet's Directory Service: Services Provided by DNS, Overview of How DNS Works, DNS Records and Messages, Peer-to-Peer Applications: P2P File Distribution, Distributed Hash Tables.
Textbook 1: chapter 2.1 to 2.6

Module -2

Transport Layer : Introduction and Transport-Layer Services: Relationship Between Transport and Network Layers, Overview of the Transport Layer in the Internet, Multiplexing and De-multiplexing: Connectionless Transport: UDP,UDP Segment Structure, UDP Checksum, Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N, Selective repeat, Connection-Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management.
Textbook 1: Chapter 3.1 to 3.5

Module -3

The Network layer: What's Inside a Router? Input Processing, Switching, Output Processing, Where Does Queuing Occur? Routing control plane, IPv6,A Brief foray into IP Security, Routing Algorithms: The Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm, Hierarchical Routing, Routing in the Internet, Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter/AS Routing: BGP, Broadcast Routing Algorithms and Multicast.
Textbook1: Ch 4: 4.3-4.7

Module -4

Computer Security Concepts: The OSI Security Architecture, Security Attacks, Security Services.
Symmetric Ciphers: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques.
Cryptosystem::Block Cipher and the Data Encryption Standard: The Data Encryption Standard, **Advanced Encryption Standard (AES)**: Finite Field Arithmetic ,AES Structure, **Public-Key Cryptography and RSA**: Principles of Public-Key Cryptosystems, The RSA Algorithm, Diffie-Hellman Key Exchange
Textbook 2: Ch 1.1 to 1.4, Ch 2.1 to 2.3, Ch 3.2, Ch 5.1 to 5.2,Ch 9.1 to 9.2,10.1

Module -5

Network access control and cloud security: IEEE 802.1X Port-Based Network Access Control ,Cloud Computing**,** Cloud Security Risks and Countermeasures, **Transport level security:** Web Security Considerations, Secure Sockets Layer, Transport Layer Security, **IP Security :**IP Security Overview
Textbook 2: Ch 16.4 to 16.5,Ch 17.1 to 17.3,Ch 20.1

PRACTICAL COMPONENT

	Part –A
1	Implement Three nodes point – to – point network with duplex links between them for different topologies. 1Set the queue size, vary the bandwidth, and find the number of packets dropped for various iterations
2	Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the throughput with respect to transmission of packets.
3	Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present and implement the above program using as message queues or FIFOs as IPC channels
4	Write a program on datagram socket for client/server to display the messages on Client side, typed at the server side.
	Part –B
5	Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion in the network.
6	Write a program to find the shortest path between vertices using bellman-ford algorithm.
7	Write a program for simple RSA algorithm to encrypt and decrypt the data
8	Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.

TEXT BOOKS:

1. James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth edition, Pearson,2017 .
2. Cryptography and Network Security: Principles and Practice, William Stallings, Pearson , Sixth Edition . (http://www.pearsonhighered.com/stallings/)

REFERENCE BOOKS:

1. Computer-Networks- Andrew S.Tanenbaum and David J.Wetherall, Pearson Education, 5thEdition. (www.pearsonhighered.com/tanenbaum)
2. Nader F Mir, Computer and Communication Networks, 2nd Edition, Pearson, 2014.

4. Syllabus Timeline

S/L	**Syllabus Timeline**	**Description**
1	Week 1-3: Application layer	**Competency:** Understand the basic functions of application protocols **Knowledge:** understand the various functions of application protocols process and service. **Skills**: know the service provided by the DNS, HTTP, FTP
2	Week 4-6: Transport layer	**Competency:** Understand the concept of transport layer and its segments and services **Knowledge:** Data volume, destination, and rate are all controlled by transport-layer protocols including TCP, UDP, DCCP, and SCTP. **Skills**: the transport layer receives the packets, sorts them, and looks for faults. Subsequently, Enable efficient network transmission,
3	Week 7-8: Network layer	**Competency:** Understand the concept of The network layer performs packetization on the data. This makes it easier to transmit the data packets in the network **Knowledge:** The main job of this layer is to maintain the quality of the data and pass and transmit it from its source to its destination. There are several important protocols that work in this layer. **Skills:** Selects the best path to transfer the data from source to its destination Knowledge

4	Week 9-10: Computer security concepts and Cryptosystem	**Competency**: Understand the concept of security attacks and services and analyse the cryptanalysis and various security attacks **Knowledge:** Mechanisms used to provide confidentiality, identity authentication, integrity authentication, source authentication, and/or support the non-repudiation of information. **Skills**: A mechanism that is designed to detect, prevent, or recover from a security attack. Security Service: A service that enhances the security of data processing systems and information transfers
5	Week 11-12: Network access control and security concepts	**Competency:** Analyses the network access control and various security concepts **Knowledge:** Unique identifying number assigned to every device connected to the internet. **Skills**: To protect network data by setting up circuits using IPsec tunnelling in which all data being sent between the two endpoints is encrypted, as with a Virtual Private Network (VPN) connection.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Utilize various teaching methods within the lecture format to network layers and security.
2	Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of network components and protocols.
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	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
6	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
7	Laboratory Learning	Utilize the facilities available in the laboratories to understand the process of network layers and protocols.

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**
	Total Marks		**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
	Total Marks		**100%**	**25**	**10**

Final CIE Marks = (A) + (B)

Semester End Examination pattern:

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 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. 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	Application layer	Students should understand the fundamental principles, components, and functions of network layers, such as protocol mechanisms, processes, and file transformation through client-server interaction.
2	Transport layers	Learning how processes are created, scheduled, transmitted, and all layers are interrelated to each other and learning each mechanism of transport layer.
3	Network layers	Learn about the network layer and how it relates to routing and security between two data sets, as well as its protocols, broadcast and multicast paths, including the Routing Algorithm, the Distance-Vector (DV) Routing Algorithm, and hierarchical routing.
4	Security attacks services and Cryptosystem	Understand security services and attacks that use data encryption and decryption techniques based on symmetric and asymmetric algorithms.
5	Network access control and Various security concepts	Analyse access control and network security, including cloud security. Transport layer security, online security, and IP security issues covered include the OSI Security Architecture, symmetric and asymmetric algorithms, and other network access and security concepts.

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

Course Outcomes (COs)

Cos	Description
M23BCS502.1	Understand and apply application layer, architecture and protocols.
M23BCS502.2	Analyse the transport layer services and UDP and TCP protocols.
M23BCS502.3	Develop the routers, IP and Routing Algorithms in network layer.
M23BCS502.4	Understand and evaluate the various security attacks, services, symmetric and asymmetric ciphers and standards.
M23BCS502.5	Design and develop the network access control and various security management.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS502.1	3	-	-	3	-	3	-	-	-	3	-	-	3	-
M23BCS502.2	-	3	-	3	-	3	-	-	-	3	-	-	-	3
M23BCS502.3	-	-	3	3	-	3	-	-	-	3	3	-	3	-
M23BCS502.4	-	-	-	3	-	3	-	-	-	3	3	-	-	3
M23BCS502.5	-	-	3	3	-	3	-	-	-	3	3	-	3	3
M23BCS502	**3**	**3**	**3**	**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**

10.Future with this Subject:

Application layer: Application layer protocols are the messaging protocols that these IoT devices used to transport data. Without application layer protocols, Internet of Things devices would have no means by which to share data and information either from device-to-device or from device-to-server.

Transport layer: Some of the transport layer devices are, Gateways: In computer networking, a gateway is a component that is part of two networks, which use different protocols. The gateway is a protocol converter which will translate one protocol into the other. The transport layer provides a total end-to-end solution for reliable communications. TCP/IP relies on the transport layer to effectively control communications between two hosts. When an IP communication session must begin or end, the transport layer is used to build this connection

Network layer: The scope of networking courses is very high and promising. The courses offered in the networking domain are the most in-demand and leading. Networking jobs are increasing, and so is the demand for Network Engineers. Dwell in the networking jobs in India and boost your career in the IT industry and emerging technologies shaping the future of networking:

5G Networks: 5G technology is the most delinquent generation of mobile communication networks, providing faster speeds and enhanced dependability compared to earlier generations.

Computer Security Concepts and Crypto-system: The future of cyber security is closely connected to quantum computing because quantum computers could change how we protect and use data. Right now, most of the ways we keep information safe in cyber security rely on the fact that some math problems are really hard for regular computers to solve.

Network access control and various security concepts: Future NAC solutions will focus on improving user experience by implementing seamless authentication mechanisms, frictionless onboarding processes, and user-friendly interfaces for policy management and self-service capabilities. The NAC can also provide endpoint security protection such as antivirus software, firewall, and vulnerability assessment with security enforcement policies and system authentication methods.

5th sem	Professional core course(PC) Software Testing	M23BIS503

1. **Prerequisites**

S/L	Proficiency	Prerequisites
01	**Basic Programming Knowledge**	**Languages**: Familiarity with at least one programming language (e.g., Python, Java, C++). **Concepts**: Understanding of core programming concepts like variables, loops, conditionals, functions, and object-oriented programming.
02	**Software Development Lifecycle (SDLC)**	**Phases**: Knowledge of various stages of SDLC (e.g., requirements analysis, design, implementation, testing, deployment, maintenance). **Models**: Understanding different development methodologies (e.g., Agile, Waterfall).
03	**Fundamentals of Software Testing**	**Types of Testing**: Understanding different types of testing (e.g., unit testing, integration testing, system testing, acceptance testing). **Testing Techniques**: Knowledge of both manual and automated testing techniques. **Test Cases**: Ability to write and execute test cases, and understand test case design techniques (e.g., equivalence partitioning, boundary value analysis).
04	**Software Quality Assurance (QA)**	**QA Principles**: Basic principles of quality assurance and its role in software development. **Metrics**: Familiarity with metrics for evaluating software quality (e.g., defect density, test coverage).
05	**Debugging Skills**	**Tools**: Ability to use debugging tools and techniques to identify and fix issues in code. **Strategies**: Understanding common debugging strategies and practices.
06	**Understanding of Software Requirements**	**Documentation**: Ability to read and interpret software requirements and specifications. **Validation**: Understanding how to ensure that software meets its requirements and user need

2. **Competencies**

S/L	Competency	KSA Description
1	**Knowledge and Understanding**	**Knowledge:** Deep understanding of software testing concepts, including various types of testing (e.g., functional, non-functional, regression, and performance testing). **Skills:** Knowledge of different testing methodologies (e.g., black-box testing, white-box testing, exploratory testing). **Attitudes:** Familiarity with the phases of SDLC and where testing fits into each phase.
2	**Test Design and Execution**	**Knowledge:** Ability to design effective test cases based on requirements and use cases. **Skills:** Skills to execute test cases manually and using automated testing tools, and to record results accurately. **Attitudes:** Proficiency in creating comprehensive test scenarios that cover different aspects of the software.
3	**Automation Skills**	**Knowledge:** Understanding of integrating automated tests into CI/CD pipelines. **Skills:** Competency in using test automation tools and frameworks (e.g., Selenium, JUnit, TestNG, Cucumber). **Attitudes:** Ability to write and maintain test scripts for automated testing.
4	**Debugging and Problem-Solving**	**Knowledge:** Ability to analyze defects, determine root causes, and provide detailed bug reports. **Skills:** Skills to identify, reproduce, and diagnose software defects **Attitudes**: Proficiency in troubleshooting and resolving complex issues.

5	**Tools and Technologies**	**Knowledge:** Proficiency in using various testing tools for different types of testing (e.g., performance testing tools like JMeter, security testing tools like OWASP ZAP). **Skills:** Familiarity with version control systems (e.g., Git) for managing test scripts and collaborating with teams.

3. Syllabus

Software Testing
SEMESTER – V

Course Code	**M23BIS503**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	4:0:0:0	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course objectives: This course will enable students to:

- Differentiate the various testing techniques
- Analyze the problem and derive suitable test cases.
- Apply suitable technique for designing of flow graph
- Explain the need for planning and monitoring a process

Module -1

Basics of Software Testing: Basic definitions, Software Quality , Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies , Levels of testing, Testing and Verification, Static Testing. **Problem Statements:** Generalized pseudocode, the triangle problem, the NextDate function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper

T1:Chapter1, T1:Chapter2.

Module -2

Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, Nextdate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. **Fault Based Testing**: Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis.

T1: Chapter 5, 6 & 7, T2: Chapter 16

Module -3

Structural Testing: Overview, Statement testing, Programme testing, Condition testing , Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice-based testing, Guidelines and observations. **Test Execution:** Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay

T1:Chapter 9 & 10, T2:Chapter 17

Module -4

Process Framework: Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties ,Analysis Testing, Improving the process, Organizational factors. **Planning and Monitoring the Process:** Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the process, the quality team **Documenting Analysis and Test:** Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.

T2: Chapter 3 & 4, T2: Chapter 20, T2: Chapter 24.

Module -5

Integration and Component-Based Software Testing: Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. Levels of Testing, **Integration Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing, A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations.

T2: Chapter 21 & 22, T1 : Chapter 12 & 13

TEXTBOOK:

1) Paul C. Jorgensen: Software Testing, A Craftsman‟s Approach, 3rd Edition, Auerbach Publications, 2008. (Listed topics only from Chapters 1, 2, 5, 6, 7, 9, 10, 12, 13)
2) Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, Wiley India, 2009. (Listed topics only from Chapters 3, 4, 16, 17, 20,21, 22,24)

REFERENCE BOOKS:

1. Software testing Principles and Practices – Gopalaswamy Ramesh, Srinivasan Desikan, 2nd Edition, Pearson, 2007
2. Software Testing – Ron Patton, 2nd edition, Pearson Education, 2004.
3. The Craft of Software Testing – Brian Marrick, Pearson Education, 1995.
4. Anirban Basu, Software Quality Assurance, Testing and Metrics, PHI, 2015.
5. Naresh Chauhan, Software Testing, Oxford University press

4.Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: **Basics of Software Testing, Problem Statements**	Basic definitions, Software Quality , Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies , Levels of testing, Testing and Verification, Static Testing.
2	Week 3-5: **Functional Testing**	Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, Nextdate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations. Fault Based Testing: Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis.
3	Week 6-8: **Structural Testing**	Overview, Statement testing, Programme testing, Condition testing , Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition-Use testing, Slice-based testing, Guidelines and observations.
4	Week 9-11: **Test Execution**	Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay
5	Week 11-13: **Process Framework Planning and Monitoring the Process**	Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties ,Analysis Testing, Improving the process, Organizational factors. Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the process, the quality team Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports
6	Week 13-15: **Integration and Component-Based Software Testing Integration Testing**	overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. Levels of Testing, Integration Testing**:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing, A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations.

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)

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

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	**Fundamentals of Software Testing**	**Understand the Purpose of Testing**: Explain why testing is essential in the software development lifecycle and how it contributes to quality assurance. **Identify Key Testing Concepts**: Define core terms such as defects, bugs, and test cases. **Differentiate Testing Types**: Describe various types of testing (e.g., unit, integration, system, acceptance)..
2	**Test Planning and Design**	**Develop Test Plans**: Create comprehensive test plans that outline the testing strategy, scope, resources, schedule, and risk management. **Design Test Cases**: Write clear and effective test cases with well-defined inputs, expected results, and execution steps. **Understand Requirements**: Analyze requirements to ensure test cases align with the specified requirements and user needs.

3	**Test Execution and Management**	**Execute Test Cases**: Perform manual and automated tests according to the test plan. **Manage Test Environments**: Set up and configure test environments to simulate real-world conditions. **Track Defects**: Report, track, and manage defects using issue-tracking tools.
4	**Test Automation**	**Understand Automation Fundamentals**: Recognize the benefits and limitations of test automation. **Use Automation Tools**: Gain hands-on experience with popular test automation tools (e.g., Selenium, JUnit, TestNG). **Develop Automated Test Scripts**: Write and maintain automated test scripts for different types of testing.
5	**Performance and Security Testing**	**Conduct Performance Testing**: Measure and analyze software performance under various conditions using tools like JMeter or LoadRunner. **Implement Security Testing**: Identify common security vulnerabilities and apply techniques to test for security weaknesses.
6	**Testing Techniques and Strategies**	**Apply Testing Techniques**: Use various testing techniques such as boundary value analysis, equivalence partitioning, and exploratory testing. **Develop Testing Strategies**: Design testing strategies tailored to different types of software and project requirements.
7	**Software Testing Lifecycle**	**Understand the Testing Lifecycle**: Learn about the various phases of the software testing lifecycle, from planning to execution and closure. **Adapt to Development Models**: Adapt testing practices to different software development methodologies, such as Agile, Waterfall, or DevOps.

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

Course Outcomes (COs)

COs	Description
M23BIS503.1	Understand the significance of software testing and quality assurance in software development
M23BIS503.2	Apply the concepts of software testing to assess the most appropriate testing method.
M23BIS503.3	Analyze the importance of testing in software development.
M23BIS503.4	Evaluate the suitable testing model to derive test cases for any given software.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BIS503.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BIS503.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
M23BIS503.3	**-**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23BIS503.4	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
M23BIS503	**3**	**2.5**	**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**	**20**		**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**	**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 software testing is evolving rapidly, driven by advancements in technology, changes in development methodologies, and the increasing complexity of software systems. Here are several key trends and potential directions for the future of software testing:

1. Increased Automation

- **Expansion of Test Automation**: As software development cycles become shorter, automation will increasingly handle repetitive and complex testing tasks. Tools like Selenium, Cypress, and newer AI-driven automation tools will become more prevalent.
- **AI and Machine Learning**: Machine learning algorithms will enhance test automation by predicting potential issues, optimizing test coverage, and generating test cases based on historical data.

2. Shift-Left Testing

- **Early Testing Integration**: Testing will move earlier in the software development lifecycle (SDLC) with practices like Test-Driven Development (TDD) and Behavior-Driven Development (BDD) becoming more common.
- **Continuous Testing**: Continuous integration and continuous deployment (CI/CD) pipelines will integrate automated testing seamlessly, ensuring that issues are detected and addressed as soon as code changes are made.

3. Increased Focus on Security

- **Shift to DevSecOps**: Security will become a more integral part of the development process with the adoption of DevSecOps practices. Automated security testing tools and techniques will be integrated into CI/CD pipelines to identify vulnerabilities early.
- **Emphasis on Threat Modeling**: More sophisticated threat modeling and security testing methods will be used to address emerging cybersecurity threats.

4. Advanced Performance Testing

- **Real-Time Performance Monitoring**: Advanced performance testing will leverage real-time monitoring and analytics to assess how software performs under different conditions and loads.
- **Synthetic and Real User Monitoring**: Combining synthetic testing (simulated user interactions) with real user monitoring (RUM) will provide a more comprehensive view of application performance and user experience.

5. Rise of AI and Intelligent Testing

- **AI-Driven Testing**: Artificial intelligence will play a significant role in creating smarter test scripts, identifying patterns in test failures, and even suggesting improvements to test cases.
- **Self-Healing Tests**: AI will enable self-healing test scripts that automatically adjust to changes in the application's UI or functionality, reducing maintenance efforts.

5th sem	Professional core course laboratory(PCL) Software Testing Lab	M23BISL504

1. Prerequisites

S/L	Proficiency	Prerequisites
01	**Hardware Requirements**	**Computers/Workstations**: Adequate number of computers with varying specifications to test software on different configurations. **Servers**: For testing server-based applications, load balancing, and stress testing. **Network Infrastructure**: Routers, switches, and cabling for connecting computers and servers. **Mobile Devices**: For testing mobile applications, you'll need a range of smartphones and tablets.
02	**Software Requirements**	**Operating Systems**: Different OS versions for compatibility testing (e.g., Windows, Linux, macOS). **Browsers**: Various web browsers for cross-browser testing (e.g., Chrome, Firefox, Edge, Safari). **Testing Tools**: **Automated Testing Tools**: Selenium, QTP, etc. **Performance Testing Tools**: JMeter, LoadRunner, etc. **Bug Tracking Tools**: Jira, Bugzilla, etc. **Version Control Systems**: Git, SVN, etc
03	**Network Configuration**	**Local Area Network (LAN)**: To connect the devices within the lab. **Internet Connectivity**: For accessing online resources, updates, and cloud-based services. **Firewall and Security**: Proper configurations to protect the lab network.
04	**Test Data**	**Sample Data**: For testing purposes, including various input scenarios, user data, and load data. **Data Management Tools**: For generating and managing test data.
05	**Documentation**	**Test Plans**: Detailed plans outlining the scope, approach, resources, and schedule for testing activities. **Test Cases**: Specific conditions under which testing will be performed. **Test Scripts**: Automated scripts used for testing.
06	**Human Resources**	**Testers/Engineers**: Skilled professionals who perform the testing. **Developers**: To fix any issues identified during testing. **System Administrators**: To manage and maintain hardware and software.
07	**Environment Setup**	**Test Environment**: A controlled setup that mimics the production environment as closely as possible. **Backup and Recovery**: Procedures and tools to ensure data integrity and recovery in case of failure.

2. Competencies

S/L	Competency	KSA Description
1	**Technical Skills**	**Knowledge:** Proficiency in using automated testing tools (e.g., Selenium, QTP) and performance testing tools (e.g., JMeter, LoadRunner). **Skills:**Ability to write and understand code in languages relevant to the application under test (e.g., Java, Python, C#). **Attitudes:** Competence in querying and managing databases (e.g., SQL, NoSQL) for data-driven testing.
2	**Testing Methodologies**	**Knowledge:** Expertise in performing manual testing, including exploratory, regression, and user acceptance testing. **Skills:** Skills in assessing system performance, load, stress, and scalability. **Attitudes:** Ability to design, implement, and maintain automated test scripts
3	**Test Planning and Design**	**Knowledge:** Developing comprehensive test strategies and plans that align with project requirements and goals. **Skills:** Ability to create detailed and effective test cases and scenarios.

		Attitudes: Identifying and managing risks associated with the testing process and product quality.
4	**Problem-Solving and Analytical Skills**	**Knowledge:** Proficiency in identifying, documenting, and communicating defects effectively. **Skills:** Skills in analyzing issues to determine their root causes and potential solutions. **Attitudes:** Ability to interpret test results and metrics to make informed decisions.
5	**Knowledge of Development Life Cycle**	**Knowledge:** Understanding of Software Development Life Cycle (SDLC) and Agile frameworks (e.g., Scrum, Kanban). **Skills:** Familiarity with Continuous Integration and Continuous Deployment (CI/CD) processes and tools.

3. Syllabus

Software Testing Lab
SEMESTER – V

Course Code	**M23BISL504**	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:

1. Explain the test cases for any given problem
2. Analyze the requirements for the given problem statement.
3. Design the solution and write test cases for the given problem.
4. Construct control flow graphs for the solution that is implemented.
5. Create appropriate document for the software artifact

l. No.	PART A – List of problems for which student should develop program and execute in theLaboratory
1	Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
2	Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.
3	Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.
4	Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, equivalence class partitioning and decision-table approach and execute the test cases and discuss the results.
5	Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
6	Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results
	PART B – Practical Based Learning
	Develop a Mini Project with documentation of suitable test-cases and their results to perform automation testing of anyE-commerce or social media web page. 03092022 Suggested Guidelines: • Create a WebDriver session.

	• Navigate to a Web page. • Locate the web elements on the navigated page. • Perform an actions on the located elements. • Assert the performed actions did the correct thing. • Report the results of the assertions. • End the session. Each inputs / data feeds (ex: website, username, password, mobile no, product name, etc.,) must be provided through a file linked with code and neither to be entered manually nor to be included in the code Use any software testing tool like selenium, Katalon, etc.,

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	Introduction, Program1
2	Week 3-4:	Program2, Program3.
3	Week 5-6:	Program4.
4	Week 7-8:	Program5.
5	Week 9-10:	Program6.
6	Week 11-12:	Part B Mini Project

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 Testing live, showing step-by-step coding and debugging.
3	Lab Exercises	Design lab exercises that require students to implement and manipulate Testing process.
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.
7	Programming Assignments	Assign programming tasks and mini projects to reinforce practical skills associated with competencies.

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):

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**

1. SEE marks for practical course shall be 50 marks
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.

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%.

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Testing Concepts**	**Comprehend the Software Testing Life Cycle (STLC)**: Gain knowledge of the phases of testing, including requirements analysis, test planning, test design, test execution, and test closure. **Learn Testing Terminology**: Familiarize with key terms and concepts such as test cases, test scripts, defects, and test metrics.
2	**Test Planning and Design**	**Develop Test Plans**: Learn to create detailed test plans that outline the scope, objectives, resources, schedule, and approach for testing activities. **Design Test Cases and Scenarios**: Acquire skills in designing effective test cases and scenarios based on requirements and specifications.
3	**Manual Testing Techniques**	**Perform Manual Testing**: Gain practical experience in executing manual test cases, including functional, exploratory, and regression testing. **Identify and Document Defects**: Learn to identify, document, and communicate defects effectively.
4	**Automated Testing**	**Understand Automated Testing**: Learn the principles and benefits of automated testing, including when and how to use it. **Create and Maintain Automated Tests**: Acquire skills in using automated testing tools (e.g., Selenium) to develop and maintain test scripts.
5	**Performance Testing**	**Conduct Performance Testing**: Learn to perform load, stress, and scalability testing using performance testing tools (e.g., JMeter). **Analyze Performance Metrics**: Gain skills in interpreting performance metrics and identifying performance bottlenecks
6	**Security Testing**	**Understand Security Testing Concepts**: Familiarize with common security vulnerabilities and testing practices. **Conduct Security Assessments**: Learn to use security testing tools and techniques to identify vulnerabilities and ensure software security.
7	**Test Environment Setup and Configuration**	**Configure Test Environments**: Gain experience in setting up and configuring test environments that mimic production environments. **Manage Test Data**: Learn to prepare, manage, and utilize test data effectively for different testing scenarios.

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

Course Outcomes (COs)

COs	Description
M23BISL504.1	Apply the suitable testing techniques for a given problem.
M23BISL504.2	Analyze the prerequisites for a given problem by writing appropriate test cases.
M23BISL504.3	Execute the test cases to solve a given problem.
M23BISL504.4	Create appropriate document for the software artifact.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BISL504.1	2	-	-	-	-	-	-	-	-	-	-	-	2	-
M23BISL504.2	-	3	-	-	-	-	-	-	-	-	-	-	-	2
M23BISL504.3	-	2	2	-	-	-	-	-	-	-	-	-	-	3
M23BISL504.4	-	-	-	2	-	-	-	-	2	-	-	-	-	-
M23BISL504	3	2.5	2	2					2				2	2.5

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Program 1-6	12.5	12.5	12.5	12.5	**50**
Total	12.5	12.5	12.5	12.5	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Program 1-6	25	25	25	25	100
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 future of software testing is evolving rapidly, driven by advancements in technology, changes in development methodologies, and the increasing complexity of software systems. Here are several key trends and potential directions for the future of software testing:

1. Increased Automation

- **Expansion of Test Automation**: As software development cycles become shorter, automation will increasingly handle repetitive and complex testing tasks. Tools like Selenium, Cypress, and newer AI-driven automation tools will become more prevalent.
- **AI and Machine Learning**: Machine learning algorithms will enhance test automation by predicting potential issues, optimizing test coverage, and generating test cases based on historical data.

2. Shift-Left Testing

- **Early Testing Integration**: Testing will move earlier in the software development lifecycle (SDLC) with practices like Test-Driven Development (TDD) and Behavior-Driven Development (BDD) becoming more common.
- **Continuous Testing**: Continuous integration and continuous deployment (CI/CD) pipelines will integrate automated testing seamlessly, ensuring that issues are detected and addressed as soon as code changes are made.

3. Increased Focus on Security

- **Shift to DevSecOps**: Security will become a more integral part of the development process with the adoption of DevSecOps practices. Automated security testing tools and techniques will be integrated into CI/CD pipelines to identify vulnerabilities early.
- **Emphasis on Threat Modeling**: More sophisticated threat modeling and security testing methods will be used to address emerging cybersecurity threats.

4. Advanced Performance Testing

- **Real-Time Performance Monitoring**: Advanced performance testing will leverage real-time monitoring and analytics to assess how software performs under different conditions and loads.
- **Synthetic and Real User Monitoring**: Combining synthetic testing (simulated user interactions) with real user monitoring (RUM) will provide a more comprehensive view of application performance and user experience.

5th Semester	**Professional Elective-I (PE)** **COMPUTER GRAPHICS**	**M23BCS505A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Skills**	• **Languages:** Proficiency in languages like C++, Python, or Java is often crucial. C++ is particularly common due to its performance and control over system resources, but Python can be useful for prototyping and scripting. • **Concepts:** Understanding basic programming concepts such as variables, control structures (loops, conditionals), functions, and data structures (arrays, lists) is essential.
2	**Mathematics**	• **Linear Algebra:** Vectors, matrices, and transformations (such as translation, rotation, scaling) are fundamental in graphics. Concepts like dot products, cross products, and matrix multiplication are regularly used. • **Geometry:** Basic geometric concepts including points, lines, and polygons. Knowledge of coordinate systems and spatial relationships is crucial. • **Calculus:** Understanding derivatives and integrals can be helpful, especially for more advanced topics like rendering equations and animations.
3	**Computer Science Fundamentals**	• **Data Structures and Algorithms:** Knowledge of data structures like arrays, linked lists, trees, and algorithms for sorting and searching can be beneficial for efficiently handling graphics data. • **Software Engineering Principles:** Understanding concepts like modularity, object-oriented design, and debugging techniques..
4	**Computer Architecture:**	• **Basic Hardware Knowledge:** Understanding how computers process data, including concepts like CPU, GPU, memory hierarchy, and parallel processing, can help in optimizing graphics performance. .
5	**Graphics-Specific Knowledge:**	• **Basic Graphics Concepts:** Familiarity with fundamental graphics concepts such as rendering pipelines, rasterization, and shading can be helpful. • **APIs and Frameworks:** Knowledge of graphics libraries and APIs like OpenGL, DirectX, or Vulkan is important for developing graphics applications. Familiarity with game engines like Unity or Unreal Engine can also be useful.
6	**Operating Systems:**	• **Basic OS Concepts:** Understanding how operating systems manage processes, memory, and file systems can aid in creating efficient graphics applications.
7	**Mathematical Software Tools:**	• **Visualization Tools:** Familiarity with software tools for visualizing mathematical concepts (e.g., MATLAB, Mathematica) can be useful, especially for complex calculations and simulations

2.Competencies

S/L	Competency	KSA Description
1	**Graphics Programming**	**Knowledge:** Proficiency in programming languages commonly used in graphics programming such as C++, Python, or Java. Understanding vectors, matrices, and transformations, and their application in graphics (e.g., modeling, transformations, and camera projections). **Skills:** Ability to write, debug, and optimize code for graphics applications. Applying mathematical algorithms to solve graphics-related problems, such as collision detection and procedural generation. **Attitudes:** Ability to think creatively and develop unique visual effects and solutions to graphics challenge
2	**Mathematics for Graphics:**	**Knowledge:**

		Knowledge of how to use vectors and matrices for transformations, such as translation, rotation, and scaling of graphical objects. Knowledge of different coordinate systems (Cartesian, polar, etc.) and their conversions. Understanding geometric transformations such as translation, rotation, scaling, and shearing. **Skills:** Skill in implementing mathematical transformations in code, such as matrix multiplication for object manipulation. Ability to apply geometric principles to model and manipulate shapes and surfaces. **Attitudes:** A willingness to continually learn and explore new mathematical techniques and advancements relevant to graphics.
3	**Rendering Techniques:**	**Knowledge:** Knowledge of how ray tracing uses geometric calculations to simulate the paths of rays of light and determine the color and brightness of pixels. Understanding how rasterization converts geometric data (e.g., triangles) into pixel data and how this process involves mathematical operations such as interpolation and scanline algorithms. **Skills:** Skills in implementing ray tracing algorithms, including calculating intersections, reflections, and refractions. Proficiency in coding rasterization algorithms, including handling edge functions and interpolating pixel values. **Attitudes:** Commitment to accurate mathematical computations and precise implementation of rendering algorithms to ensure high-quality results. A dedication to refining rendering techniques to achieve the best possible visual fidelity and performance.
4	**Data Structures and Algorithms:**	**Knowledge:** Knowledge of how arrays,linked list, stacks and queues work, their indexing, and their applications. Knowledge of working of heap, graphs and trees **Skills:** Proficiency in implementing various algorithms and data structures in programming languages such as C++, Java, or Python. Skills in optimizing algorithms for efficiency in terms of time and space. **Attitudes:** Ability to critically analyze problems, identify suitable data structures and algorithms, and evaluate their effectiveness. Precision in implementing and testing algorithms to ensure accuracy and efficiency.
5	**3D Modeling and Animation:**	**Knowledge:** Knowledge of fundamental modeling techniques such as transformations, leveling, and edge looping. Awareness of good topology practices to ensure models are clean, manageable, and efficient for animation. **Skills:** Ability to create accurate and detailed 3D models using various techniques and tools. Skills in optimizing 3D models for performance, including reducing polygon count and managing textures efficiently. **Attitudes:** Willingness to explore creative ideas and approaches in modeling and animation. An eye for aesthetics and attention to detail in creating visually appealing models and animations.
6	**Software and Tools:**	**Knowledge:** Knowledge of tools like Adobe Photoshop, CorelDRAW for creating and editing 2D images and textures.

		Skills: Understanding of software such as spyder ,openGL API Skills in navigating and utilizing the features and functionalities of graphics software effectively. **Attitudes:** A willingness to continuously learn about new tools, updates, and techniques in the field of computer graphics.

3. Syllabus

COMPUTER GRAPHICS
SEMESTER – V

Course Code	**M23BCS505A**	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:

1.To learn hardware, software and OpenGL Graphics Primitives
2.To analyze interactive computer graphics using OpenGL
3. To design and implement algorithms for 2D graphics Primitives and attributes.
4. To Demonstrate Geometric transformations, viewing on both 2D and 3D objects
5. To Infer the representation of curves, surfaces, color and Illumination models.

Module -1

Computer Graphics and OpenGL:

Computer Graphics: Basics of computer graphics, Application of Computer Graphics,Random Scan and Raster Scan displays, graphics software. OpenGL: Introduction to OpenGL ,coordinate reference frames, specifying two-dimensional world coordinate reference frames in OpenGL, OpenGL point functions, OpenGL line functions, point attributes, line attributes, OpenGL point attribute functions, OpenGL line attribute functions.

Text Book -1:Chapter -1,2,3;

Module -2

Line drawing and Circle drawing algorithm. and 2D & 3D Geometric Transformation:

Line drawing algorithms(DDA, Bresenham‟s), circle generation algorithms (Bresenham‟s).

2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2DComposite transformations, other 2D transformations. OpenGL geometric transformations function, Transformations: 3D translation, rotation, scaling, composite 3D transformations. **OpenGL** geometric transformations functions.

Text Book -1:Chapter -5,6,8;

Module -3

2D viewing & Clipping:

2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions. Clipping: clipping window, normalization and viewport transformations, clipping algorithms,2D point clipping, 2D line clipping algorithms: cohen-sutherland line clipping only -polygon fill area clipping: Sutherland-Hodgeman polygon clipping algorithm only.

Text Book -1:Chapter -7;

Module -4

3D Viewing:

3DViewing:3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters , Transformation from world to viewing coordinates, Projection transformation, orthogonal projections, perspective projections, The viewport transformation and 3D screen coordinates. OpenGL 3D viewing functions..

Text Book -1:Chapter -9;

Module -5

Input & interaction & Color and Illumination Models:

Input and Interaction: Input devices, Display Lists, Display Lists and Modeling, Programming Event Driven Input, Menus Picking, Logic operations.

Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong model, Corresponding openGL functions.

Text Book -1:Chapter -12, Text Book2- Chapter 3;

TEXTBOOKS:

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011

2. 2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008

REFERENCE BOOKS:

1. James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education
2. 2. Xiang, Plastock : Computer Graphics , sham‟s outline series, 2nd edition, TMG

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: **Computer Graphics and OpenGL**	• Understanding basics of computer graphics, Application of Computer Graphics: Random Scan and Raster Scan displays, graphics software. • OpenGL: Introduction to OpenGL , OpenGL point functions, OpenGL line functions, OpenGL line attribute functions,
2	Week 4-5: **Line drawing and Circle drawing algorithm. and 2D & 3D Geometric Transformation:**	• Line drawing algorithms(DDA, Bresenham‟s), circle generation algorithms (Bresenham‟s) • Understanding 2DGeometric Transformations: Basic 2D Geometric Transformations, 2DComposite transformations, 3D Geometric Transformation.
3	Week 6-8: **2D viewing & Clipping:**	• Understanding Clipping: clipping window, clipping algorithms,2D point clipping, • 2D line clipping algorithms: cohen-sutherland line clipping only -polygon fill area clipping: • Sutherland-Hodgeman polygon clipping algorithm. s. .
4	Week 9-10: **3D viewing**	• Understanding 3DViewing:3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters , • OpenGL 3D viewing functions. ,Visible Surface Detection Methods: Classification of visible surface Detection algorithms..
5	Week 11-13 **Input and interaction, Colour and Illumiantion models**	• Input and Interaction: Input devices, clients and servers, Display Lists, • Building Interactive Models, • Understanding about colour and illumination model.

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 the concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
5	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
6	Laboratory Learning	Utilize the facilities available in the laboratories to understand the behavior of the materials by performing few experiments.

5. 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

	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	**Basics of computer graphics**	learn about the evolution of computer graphics, different types of graphics (2D and 3D), and key terms such as pixels, resolution, and color models.
2	**2D geometric transformations and viewing**	Learn about fundamental transformations such as translation, scaling, and rotation. Understand how these operations affect the position, size, and orientation of geometric objects.
3	**Clipping and clipping windows**	Learn why clipping is necessary for rendering efficiency and visual accuracy. Study how clipping windows define the region of interest in world coordinates and how viewports map these regions to screen coordinates.
4	**3D geometric transformations**	Gain a foundational understanding of basic 3D transformations and their effects on objects. Learn about fundamental transformations in 3D space, including translation, rotation, and scaling. Understand how these transformations affect the position, orientation, and size of 3D objects.
5	**3d viewing**	Gain a foundational understanding of 3D viewing and how it affects rendering. Learn about the concept of the camera or view frustum, which defines the 3D space visible to the user. Understand the role of the viewing volume in determining what is rendered.
6	**Understand the Fundamentals of Animation**	Gain a foundational understanding of key principles and concepts in computer animation. Learn about the basic principles of animation such as timing, spacing, squash and stretch, anticipation, and follow-through. Understand how these principles contribute to creating realistic and engaging animations

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

Cos	Description
M23BCS505A.1	**Understand and Apply** fundamental concepts of computer graphics 2D graphics primitives and attributes
M23BCS505A.2	**Analyze** different Geometric transformations on both 2D and 3D objects.
M23BCS505A.3	**Design** and implement algorithms of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
M23BCS505A.4	**Develop** programs and graphics packages using OpenGL,suitable hardware and software**.**

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS505A.1	3		-	-	-	-	-	-	-	-	-	-	3	-
M23BCS505A.2	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCS505A.3	-	-	3		-	-	-	-	-	-	-	-	3	-
M23BCS505A.4	-	-	-	3	-	-	-	-	-	-	-	-	-	3
M23BCS505A	**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				**10**	**10**
Total	**10**	**10**	**10**	**20**	**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				**20**	**20**
Total	**20**	**20**	**20**	**40**	**100**

10. **Future with this Subject:**

- Real-Time RayTracing -Real-time ray tracing, enabled by advancements in GPU technology, allows for highly realistic rendering by simulating the behavior of light in a scene. This technique produces high-quality reflections, refractions, and shadows.
- E**nhanced Virtual and Augmented Reality (VR/AR)**-VR and AR technologies are evolving rapidly, offering more immersive and interactive experiences. Advancements in graphics technology will improve the visual fidelity and responsiveness of VR and AR applications.
- **AI and Machine Learning Integration**-Artificial intelligence (AI) and machine learning are increasingly being integrated into graphics processes. These technologies can enhance image generation, improve rendering efficiency, and automate tasks such as texture creation and character animation
- **.High Dynamic Range (HDR) and Wide Color Gamut** : HDR and wide color gamut technologies enhance the range of brightness and color accuracy in graphics. As displays and content production tools adopt these technologies, visual quality will improve significant
- **Advanced Graphics Processing Units (GPUs)-**Future GPUs will continue to push the boundaries of performance and efficiency. Innovations in GPU architecture, memory, and parallel processing will drive advancements in graphics rendering and computation

5th Semester	**Professional Elective-I (PE)** **CLOUD COMPUTING**	**M23BCS505B**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Students should have a solid understanding of programming concepts, particularly in Python, as it is widely used in data science for data manipulation, analysis, and machine learning. Knowledge of data structures (e.g., lists, dictionaries) and control structures (e.g., loops, conditionals) is essential.
2	**Mathematics:**	Understanding of solving linear and quadratic equations Proficiency in differential and integral calculus, including applications. Familiarity with geometric shapes, angles, trigonometric functions, and their properties.
3	**Computer Networks and Security**	Computer Networks, Network Security Understanding of network protocols, TCP/IP, DNS, and basic network configurations. Information Security, Cryptography
4	**Operating Systems**	Operating Systems, Systems Programming. Knowledge of process management, memory management, file systems, and system calls.
5	**Fundamentals of Probability**	Knowledge of probability theory, including conditional probability, Bayes' theorem, and probability distributions, is essential. This will help students understand the statistical underpinnings of many data science algorithms, such as Naive Bayes

2. Competencies

S/L	Competency	KSA Description
1	Introduction to cloud computing	**Knowledge:** Understanding the definition, scope, and significance of Cloud computing. Awareness of the Historical Developments, Building Cloud Computing Environments, Amazon Web Services (AWS), Google App Engine. **Skills:** Ability to articulate the role and impact of Cloud in various industries. Skill in identifying the different components of the Cloud workflow. **Attitudes:** Curiosity about the evolving field of Cloud and its applications. Appreciation for the interdisciplinary nature of Cloud, integrating statistics, mathematics, and domain knowledge.
2	Virtualization	**Knowledge:** Understanding the Cloud computing that enables the creation and management of virtual instances of physical resources, such as servers, storage, and networks. **Skills:** Problem-solving and project management capabilities **.** **Attitudes:** The right mindset can significantly impact how effectively one can implement, manage, and optimize virtualized environments
3	Cloud Computing Architecture	**Knowledge:** Understanding the Detailed overview of the knowledge components related to cloud computing architecture ub-components required for cloud computing. These components typically consist of a front-end platform (client or device), back-end platforms (servers, storage), a cloud-based delivery, and a network (usually the internet). **Skills:** Blend of technical skills, strategic thinking, and an understanding of the underlying principles and best practices, Cloud Service Model Expertise, Virtualization and Networking **Attitudes:**

		Play a crucial role in the effective design, implementation, and management of cloud computing architecture.
4	Cloud Platforms in Industry	**Knowledge:** Understanding of Compute services, Storage services, Communication services, Additional services. Google Architecture and core concepts, Application life cycle. **Skills:** Edge Computing , Architectural Design, Data Management and Analytics and Troubleshooting and Support **Attitudes:** Attention to detail in cloud platforms
5	Cloud Applications	**Knowledge:** Competency-based knowledge for cloud applications typically encompasses a variety of skills across multiple domains, including infrastructure, development. **Skills:** Cloud applications are focused on technical proficiency, problem-solving, and the ability to leverage cloud platforms to create, deploy, and manage applications efficiently. **Attitudes:** Attention to detail in cloud Application.

3. Syllabus

<table>
<tr><td colspan="4"><b>CLOUD COMPUTING
SEMESTER – V</b></td></tr>
<tr><td>Course Code</td><td><b>M23BCS505B</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>
1. Compare cloud computing environment utilized for real time applications.
2. Identify various models of cloud computing.
3. Analyze how to design cloud native applications.
4. Examine the importance of Cloud Virtualization Technologies.</td></tr>
<tr><td colspan="4"><b>Module -1</b></td></tr>
<tr><td colspan="4">Introduction ,Cloud Computing at a Glance, Historical Developments, Building Cloud Computing Environments, Amazon Web Services (AWS), Google App Engine, Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka
<b>Textbook 1: Chapter 1: 1.1,1.2 and 1.3</b></td></tr>
<tr><td colspan="4"><b>Module -2</b></td></tr>
<tr><td colspan="4"><b>Virtualization:</b> Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples.
<b>Textbook 1 : Chapter 3: 3.1 to 3.6</b></td></tr>
<tr><td colspan="4"><b>Module -3</b></td></tr>
<tr><td colspan="4"><b>Cloud Computing Architecture:</b> Introduction, Cloud Reference Model, Types of Clouds, Economics of the Cloud, Open Challenges.
<b>Textbook 1: Chapter 4: 4.1 to 4.5</b></td></tr>
<tr><td colspan="4"><b>Module -4</b></td></tr>
<tr><td colspan="4"><b>Cloud Platforms in Industry</b>
Amazon web services: - Compute services, Storage services, Communication services, Additional services. Google AppEngine: - Architecture and core concepts, Application life cycle, Cost model, Observations.
<b>Textbook 1: Chapter 9: 9.1 to 9.2</b></td></tr>
<tr><td colspan="4"><b>Module -5</b></td></tr>
<tr><td colspan="4"><b>Cloud Applications</b>
Scientific applications: - HealthCare: ECG analysis in the cloud, Biology: gene expression data analysis for cancer diagnosis, Geoscience: satellite image processing. Business and consumer applications: CRM and ERP, Social networking, media applications.
<b>Textbook 1: Chapter 10: 10.1 to 10.2</b></td></tr>
</table>

Textbooks

1. Rajkumar Buyya, Christian Vecchiola, and Thamrai Selvi Mastering Cloud Computing McGraw Hill Education.
2. Toby Velte, Anthony Velte, Cloud Computing: A Practical Approach, McGraw-Hill Osborne Media.

Reference Books

1. Borko Furht. Armando Escalante, "Handbook of Cloud Computing", Springer
2. George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Publication.

Weblinks and Video Lectures (e-Resources):

https://www.youtube.com/watch?v=1N3oqYhzHv4
https://www.youtube.com/watch?v=RWgW-CgdIk0

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3:	Introduction ,Cloud Computing at a Glance, Historical Developments, Building Cloud Computing Environments, Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop, Force.com and Salesforce.com
2	Week 4-6:	**Virtualization:** Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization
3	Week 8-11:	**Cloud Computing Architecture:** Introduction, Cloud Reference Model, Types of Clouds, Economics of the Cloud, Open Challenges
4	Week 7-8:	**Cloud Platforms in Industry** Amazon web services: - Compute services, Storage services, Communication services, Additional services. Google AppEngine: - Architecture and core concepts, Application life cycle, Cost model.
5	Week 9-12:	**Cloud Applications** Scientific applications: - HealthCare: ECG analysis in the cloud, Biology: gene expression data analysis for cancer diagnosis, Geoscience: satellite image processing. Business and consumer applications: CRM and ERP, Social networking, media applications.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lectures and Interactive Discussions	Provide clear, concise explanations of key concepts, theories, and algorithms in each module. Use visual aids, such as slides and diagrams, to enhance understanding.
2	Case Studies and Real-World Applications	Incorporate visual aids like videos/animations to enhance understanding the concepts. Incorporate case studies like the Real Direct example in Week 4 to demonstrate the application of data science concepts in real-world scenarios. This helps students see the relevance of what they are learning.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Project-Based Learning	Organize students into small groups to discuss complex topics, such as the ethical implications of data science
5	Lectures and Interactive Discussions	Provide clear, concise explanations of key concepts, theories, and algorithms in each module. Use visual aids, such as slides and diagrams, to enhance understanding.

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

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	Introduction to Cloud Computing	Students will be able to define cloud computing and explain its significance in the context Cloud Computing Environments.
2	Virtualization	Students will be able to know the Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization.
3	Cloud Computing Architecture:	Cloud Reference Model, Types of Clouds, Economics of the Cloud, Open Challenges.
4	Cloud Platforms in Industry	Amazon web services: - Compute services, Storage services, Communication services, Additional services. Google AppEngine: - Architecture and core concepts, Application life cycle, Cost model
5	Cloud Applications	Students will be able know cloud application like Health ECG analysis in the cloud, and gene expression data analysis for cancer diagnosis.

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

Course Outcomes (COs)

Cos	Description
M23BCS505B.1	Compare cloud computing environment utilized for real time applications.
M23BCS505B.2	Identify various models of cloud computing.
M23BCS505B.3	Analyze how to design cloud native applications.
M23BCS505B.4	Examine the importance of Cloud Virtualization Technologies.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS505B.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS505B.2	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS505B.3	3	3	-	3	-	3	-	-	-	-	-	3	3	3
M23BCS505B.4	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS505B	**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				**10**	**10**
Total	**10**	**10**	**10**	**20**	**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				**20**	**20**
Total	**20**	**20**	**20**	**20**	**100**

10. Future with this Subject:

- **Serverless Architectures**: Simplifying application deployment and scaling by abstracting server management. This includes further advancements in Function as a Service (FaaS) and Backend as a Service (BaaS).
- **Edge Computing**: Expanding cloud capabilities to the edge of the network to reduce latency and improve performance, particularly for IoT and real-time applications.
- **Artificial Intelligence and Machine Learning**: Integrating AI and ML with cloud services to enable more intelligent automation, enhanced analytics, and better decision-making capabilities
- **Security and Privacy**: Advancing encryption, identity management, and threat detection to address growing concerns about data protection and compliance in cloud environments.
- **Quantum Computing**: Exploring how quantum computing can be integrated into cloud platforms to tackle complex problems beyond the capabilities of classical computers.
- **Advanced Networking**: Developing new networking technologies and protocols to support the growing demands of cloud computing, including better support for high-speed data transfer and network slice

5th Semester	**Professional Elective I (PE)** **DATA MINING AND DATA WAREHOUSING**	**M23BIS505C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Statistical Knowledge**:	Understanding statistical concepts such as distributions, hypothesis testing, correlation, and regression is crucial. This helps in making sense of the data and drawing meaningful conclusions.
2	Programming Skills	Proficiency in programming languages commonly used in data mining, such as Python or R, is important. These languages have extensive libraries and frameworks for data analysis
3	**Data Management**	Knowledge of database management systems (DBMS) and SQL (Structured Query Language) is necessary for querying and manipulating data.
4	**Machine Learning**	Understanding basic machine learning concepts and algorithms, such as classification, clustering, and association rules, is important. Data mining often involves applying these techniques to extract patterns from data.
5	**Data Visualization**	Skills in data visualization tools and techniques help in interpreting and presenting the results of data mining efforts effectively
6	**Database Fundamentals**	Understanding relational database principles, including normalization, indexing, and schema design, is crucial. Data warehousing relies on structured data storage and retrieval. **SQL**: Proficiency in SQL is essential for querying data and managing database structures within a data warehouse.
7	**ETL Processes**	Knowledge of Extract, Transform, Load (ETL) processes is important for moving data from source systems to the data warehouse. This includes data cleansing, transformation, and loading techniques.
8	**Data Modeling**	Understanding data modeling concepts such as star schema, snowflake schema, and fact tables is important for designing efficient data warehouses.
9	**Business Intelligence (BI)**:	Familiarity with BI concepts and tools helps in utilizing data warehouses effectively for reporting and analysis.
10	**Data Warehousing Technologies**:	Familiarity with data warehousing solutions and technologies such as Oracle, Microsoft SQL Server, IBM Db2, or cloud-based solutions like Amazon Redshift, Google Big Query, and Snowflake can be helpful.

2. Competencies

S/L	Competency	KSA Description
1	**Statistical and Analytical Skills**:	**Knowledge:** Ability to apply statistical methods to interpret data and validate results. **Skills:** Proficiency in testing hypotheses to draw conclusions from data. **Attitudes:** Expertise in designing data models, including star and snowflake schemas, to structure data for efficient querying and reporting..
2	**Machine Learning and Algorithm Knowledge**:	**Knowledge:** Knowledge of various algorithms such as decision trees, neural networks, support vector machines, clustering algorithms (like k-means), and association rule mining. **Skills:** Skills in evaluating and validating models using metrics like accuracy, precision, recall, F1 score, and ROC-AUC. **Attitudes:** Ability to write scripts for data processing and automation of repetitive tasks.
3	**Data Management / Data Visualization**	**Knowledge:** Competence in preprocessing and cleaning data to ensure quality and consistency. **Skills:** Skills in transforming raw data into a usable format for analysis. **Attitudes:** Ability to present data insights clearly and persuasively to stakeholders.
4	**ETL Processes / SQL**	**Knowledge:** Proficiency with ETL tools such as Informatica, Talend, or Microsoft SSIS for extracting, transforming, and loading data. **Skills:** Skills in integrating data from various sources into the data warehouse.

		Attitudes: Ability to optimize queries and database performance..
5	**Data Warehousing Technologies / Business Intelligence (BI)**:	**Knowledge:** Understanding of architecture principles, including data marts, staging areas, and OLAP cubes. **Skills:** Skills in creating interactive dashboards that provide actionable insights. **Attitudes:** Ability to align data warehousing solutions with business goals and strategy.

3.Syllabus

DATA MINING AND DATA WAREHOUSING
SEMESTER – V

Course Code	**M23BIS505C**	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 and implement classical models and algorithms in data warehouses and data mining.
- Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- Master data mining techniques in various applications like social, scientific and environmental context.
- Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

MODULE-1 (8 Hours)

Data Warehouse: Introduction to Data Ware House, Differences between operational data base systems and data Ware House, Data Ware House characteristics, Data Ware House Architecture and its components, Extraction-Transformation-Loading, Logical (Multi-Dimensional), Data Modeling, Schema Design, star and snow-Flake Schema, Fact Constellation, Fact Table, Fully Addictive, Semi-Addictive, Non-Addictive Measures; FactLess-Facts, Dimension Table characteristics; Fact-Less-Facts, Dimension Table characteristics; OLAP cube, OLAP Operations, OLAP Server Architecture-ROLAP, MOLAP and HOLAP.

MODULE-2 (8 Hours)

Introduction to Data Mining: Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing- Data Cleaning, Missing Data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binaryzation , Data Transformation; Measures of similarity and dissimilarity-Basics

MODULE-3 (8 Hours)

Association Rules: Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation, APRIORI Algorithm, The Partition Algorithms, FP-Growth Algorithms, Compact Representation of Frequent Item Set-Maximal Frequent Item Set, Closed Frequent Item Set

MODULE-4 (8 Hours)

Classification: Problem definition, General Approaches to solving a classification problem, Evaluation of Classifiers, Classification techniques, Decision trees-Decision Tree Construction, Methods for expressing attribute test conditions, Measures for Selecting the Best split, Algorithm for Decision tree Induction, Naïve-Bayes Classifier, Bayesian Belief Networks; K-nearest neighbor classification-Algorithm and characteristics.

MODULE-5 (8 Hours)

Clustering: Problem Definition, Clustering overview, Evaluation of clustering algorithms, Partitioning clustering K-Means Algorithm, K-Means Additional Issues, PAM Algorithm, Hierarchical Clustering-Algorithm- Agglomerative Methods and Divisive Methods, Basic Agglomerative Hierarchical Clustering Algorithm, Specific techniques, Key Issues in Hierarchical Clustering, Strengths and weakness, Outlier Detection

Text Books:

1. Data Mining-Concepts and Techniques- Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, 2 Edition, 2006.
2. Introduction to Data Mining, Pang-Ning Tan, Vipin Kumar, Michael Steinbanch, Pearson Education

REFERENCES:

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.

2. Data Ware Housing Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.
3. The Data Ware House Life Cycle Toolkit- Ralph Kimball, Wiley Student Edition.
4. Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: **Data Warehouse**	Introduction to Data Ware House, Differences between operational data base systems and data Ware House, Data Ware House characteristics, Data Ware House Architecture and its components, Extraction-Transformation-Loading, Logical (Multi-Dimensional), Data Modeling, Schema Design, star and snow-Flake Schema, Fact Constellation, Fact Table, Fully Addictive, Semi-Addictive, Non-Addictive Measures; FactLess-Facts, Dimension Table characteristics.
2	Week 3-4: **Data Warehouse and Introduction to Data Mining**	OLAP cube, OLAP Operations, OLAP Server Architecture-ROLAP, MOLAP and HOLAP. Introduction, What is Data Mining, Definition, KDD, Challenges, Data Mining Tasks, Data Preprocessing- Data Cleaning, Missing Data, Dimensionality Reduction, Feature Subset Selection, Discretization and Binaryzation.
3	Week 5-6: **Introduction to Data Mining and Association Rules**	Data Transformation; Measures of similarity and Dissimilarity-Basics. Problem Definition, Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation, APRIORI Algorithm, The Partition Algorithms, FP-Growth Algorithms
4	Week 7-8: **Association Rules and Classification**	Compact Representation of Frequent Item Set-Maximal Frequent Item Set, Closed Frequent Item Set. Problem definition, General Approaches to solving a classification problem, Evaluation of Classifiers, Classification techniques, Decision trees-Decision Tree Construction, Methods for expressing attribute test conditions, Measures for Selecting the Best split, Algorithm for Decision tree Induction
5	Week 9-10: **Classification and Clustering**	Naïve-Bayes Classifier, Bayesian Belief Networks; K-nearest neighbor Classification-Algorithm and characteristics. Problem Definition, Clustering overview, Evaluation of clustering algorithms, Partitioning clustering K-Means Algorithm, K-Means Additional Issues, PAM Algorithm,
6	Week 11-12: **Clustering**	Hierarchical Clustering-Algorithm- Agglomerative Methods and Divisive Methods, Basic Agglomerative Hierarchical Clustering Algorithm, Specific techniques, Key Issues in Hierarchical Clustering, Strengths and weakness, Outlier Detection

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Conceptual Framework**	Start with the basics of statistics, probability, and machine learning. Explain core concepts such as supervised vs. unsupervised learning, clustering, classification, and association rules. Begin with a solid understanding of database management principles, data modeling, and schema design.
2	**Practical Hands-On Training**	Introduce tools and software commonly used in data mining such as Python (with libraries like scikit-learn, pandas), R, Weka, or RapidMiner. Provide training on ETL tools like Talend, Informatica, or SSIS, including how to design and implement ETL workflows

3	**Project-Based Learning**	Assign projects that require students to perform data mining tasks end-to-end, from data cleaning to model deployment. This could involve working on data sets from various domains. Work on projects that involve integrating data from multiple sources and creating a cohesive data warehouse solution.
4	**Interactive Learning**	Use simulations or case studies to demonstrate the application of data mining techniques in different scenarios. Organize workshops and guest lectures by industry experts to expose students to current trends and practices. Use case studies to explore real-world data warehousing challenges and solutions. Analyze case studies from different industries to understand diverse applications.
5	**Data Visualization**	Teach how to use data visualization tools like Tableau or Matplotlib to present data mining results.
6	**Assessments and Evaluation**	Assess students on their ability to design data models and implement ETL processes.
7	**Blended Learning**	**Online Resources**: Utilize online tutorials, videos, and MOOCs to supplement classroom learning and provide flexible study options. **Interactive Platforms**: Encourage the use of interactive platforms and coding exercises (e.g., Jupyter Notebooks) for hands-on practice.
8	**Collaborative Learning**	**Study Groups**: Promote the formation of study groups for collaborative problem-solving and peer learning. **Discussion Forums**: Use online forums or classroom discussions to explore concepts and address questions.

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

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 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 Fundamental Concepts**	Understand key concepts such as data pre-processing, pattern discovery, classification, clustering, association rules, and anomaly detection. Understand fundamental concepts such as data warehousing architecture, data marts, OLAP vs. OLTP systems, and star/snowflake schemas
2	**Data Preparation and Cleaning**	Learn how to preprocess data by handling missing values, normalization, and transformation. Develop skills in cleaning and preparing data to ensure quality and consistency before analysis.

3	**Application of Data Mining Algorithms**	Gain proficiency in implementing and tuning various data mining algorithms, such as decision trees, k-means clustering, and association rule mining.
4	**Problem Solving and Critical Thinking**	Apply data mining techniques to solve complex problems and make data-driven decisions. Critically evaluate the effectiveness of different data mining approaches and their impact on business or research outcomes.
5	**ETL Processes**	Understand the Extract, Transform, Load (ETL) process, including data extraction, transformation, and loading techniques.
6	**Data Integration and Quality**	Understand how to integrate data from multiple sources into a unified data warehouse. Learn to ensure data quality and integrity within the data warehouse through proper governance and cleansing practices.
7	**Integration of Data Mining and Data Warehousing**	**Holistic Understanding**: Develop an understanding of how data mining and data warehousing complement each other in the data lifecycle. **Integration Skills**: Learn to integrate data mining techniques with data warehousing systems to enhance data-driven insights.
8	**Ethical and Legal Considerations**	Be aware of ethical considerations and legal requirements related to data privacy, security, and compliance in both data mining and data warehousing contexts.

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

COs	Description
M23BIS505C.1	Understand the functionality of the various data mining and data warehousing component
M23BIS505C.2	Apply the strengths and limitations of various data mining and data warehousing models
M23BIS505C.3	Analyze the different methodologies used in data mining and data ware housing.
M23BIS505C.4	Evaluate the different approaches of data ware housing and data mining with various technologies.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BIS505C.1	**3**												**2**	
M23BIS505C.2		**3**												**2**
M23BIS505C.3			**3**										**2**	
M23BIS505C.4				**3**									**2**	
M23BIS505C	**3**	**3**	**3**	**3**									**2**	**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				**10**	**10**
Total	**15**	**15**	**10**	**20**	**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				**20**	**20**
Total	**30**	**30**	**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.**

The future of data mining and data warehousing looks incredibly promising and dynamic, driven by several key trends and advancements. Here are some aspects to consider,

1. Increased Integration with AI and Machine Learning

Predictive Analytics: Data mining will increasingly leverage machine learning algorithms to provide more accurate predictions and insights. AI can enhance the extraction of patterns from complex data sets, making forecasts and recommendations more reliable.

2. Enhanced Data Warehousing Technologies

Cloud Data Warehousing: The shift to cloud-based data warehousing solutions will continue, offering scalability, flexibility, and cost-effectiveness. Major providers like Amazon Redshift, Google BigQuery, and Snowflake are leading this transformation.

3. Real-Time Data Processing

Streaming Analytics: The ability to analyze and act on data in real-time will become more prevalent. Technologies like Apache Kafka and Apache Flink are leading this space, enabling businesses to respond instantly to emerging trends and anomalies.

Edge Computing: Processing data closer to where it is generated (at the edge) will reduce latency and bandwidth issues, making real-time data analysis more efficient.

4. Data Privacy and Security

Enhanced Security Measures: With growing concerns over data breaches, future data warehousing solutions will incorporate more advanced security features, including encryption, access controls, and anomaly detection.

5. Big Data and Scalability

Handling Larger Volumes: As data continues to grow exponentially, data warehousing solutions will evolve to handle larger volumes of data more efficiently. Technologies like distributed databases and data lakes will play a crucial role.

5th Semester	**Professional Elective-I (PE) ADVANCED JAVA**	**M23BCS505D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Knowledge of Programming**:	• Understanding of programming fundamentals (variables, control structures, loops, functions).
2	**Solid Grasp of Core Java**:	• Mastery of Java basics including syntax, data types, operators, and basic OOP concepts.
3	**Experience with Java Development Environment**:	• Proficiency in using an IDE (like IntelliJ IDEA, Eclipse, or NetBeans). • Familiarity with Java development tools (javac, java, jar).
4	**Basic Understanding of Software Development Life Cycle:**	• Knowledge of the phases of software development, from requirement gathering to deployment.

2. Competencies

S/L	Competency	KSA Description
1	**Enumerations, Autoboxing, and Annotations**	Knowledge: • Understand the fundamentals of Enumerations in Java, including the purpose and usage of values() and valueOf() methods. • Recognize that Java Enumerations are class types and that they inherit from Enum. • Grasp the concept of Autoboxing and Unboxing in Java, particularly how they occur in expressions and the prevention of errors related to primitive and wrapper types. • Understand the basics of Annotations in Java, including retention policies, obtaining annotations at runtime using reflection, and different types of annotations like Marker, Single-member, and Built-in annotations. Skills: • Ability to implement and utilize enumerations in Java applications. • Proficient in applying Autoboxing and Unboxing in code, particularly in complex expressions and boolean/character values. • Ability to create and apply custom annotations, specify retention policies, and retrieve annotation data at runtime using reflection. • Effectively handle and interpret annotated elements within Java applications. Attitudes: • Develop a cautious approach to using Autoboxing and Unboxing, understanding the potential for performance issues or unexpected behavior. • Foster a mindset that values the importance of code readability and maintainability when using annotations and enumerations. • Embrace best practices for using Java annotations to document code and make it more understandable for future developers.
2	**Generics**	Knowledge: • Understand the concept of Generics in Java, including the use of generic classes, methods, and interfaces. • Familiarize with bounded types, wildcard arguments, and bounded wildcards in generics. • Recognize the significance of type erasure and how it affects generics. • Comprehend the limitations and restrictions associated with generics, including ambiguity errors and compatibility with legacy code. Skills: • Ability to define and implement generic classes, methods, and interfaces with one or more type parameters. • Competent in using bounded types and wildcards to create flexible and type-safe code.

		• Ability to troubleshoot and resolve issues related to generic type erasure and ambiguity errors. • Capable of integrating generics into existing legacy code without introducing compatibility issues. Attitudes: • Develop a careful and thoughtful approach to using generics, prioritizing type safety and code reusability. • Value the importance of understanding the underlying mechanics of generics, such as type erasure, to avoid common pitfalls. • Embrace a mindset that encourages the use of generics to write cleaner, more maintainable, and robust code.
3	**String Handling**	Knowledge: • Understand the different constructors available for creating String objects and how to determine the length of a string. • Familiarize with special string operations, character extraction, comparison, and searching within strings. • Grasp the methods available for modifying strings, converting data types using valueOf(), and changing the case of characters. • Recognize the differences between String, StringBuffer, and StringBuilder, including their use cases. Skills: • Ability to perform various string operations, including comparison, extraction, searching, and modification, in Java. • Proficient in using String Buffer and String Builder for efficient string manipulation in performance-sensitive applications. • Competent in applying data conversion techniques using valueOf() and case conversion methods. Attitudes: • Develop an appreciation for the importance of efficient string handling, particularly in performance-critical applications. • Foster a detail-oriented mindset when working with strings to avoid common errors such as Null Pointer Exception or incorrect string manipulation. • Embrace best practices for choosing between String, String Buffer, and String Builder based on the specific needs of the application.
4	**Servlets**	Knowledge: • Understand the life cycle of a servlet, including the initialization, service, and destruction phases. • Familiarize with the Servlet API, including the javax.servlet and javax.servlet.http packages. • Grasp the methods for handling HTTP requests and responses, using cookies, and managing session tracking in servlets. • Recognize the basic structure and functionality of Java Server Pages (JSP), including JSP tags, variables, control statements, and session management. Skills: • Ability to develop, deploy, and manage servlets in a Java web application. • Proficient in handling HTTP requests/responses, managing sessions, and using cookies within servlets. • Competent in integrating JSPs with servlets for dynamic web content generation. • Capable of managing user sessions, parsing request data, and maintaining state across requests using session objects and cookies. Attitudes: • Develop a user-centric approach to servlet development, ensuring efficient handling of HTTP requests and responsive web applications.

		• Foster a security-conscious mindset, particularly regarding session management and the use of cookies in web applications. • Embrace the principles of good web application design, prioritizing scalability, maintainability, and performance.
5	**JDBC**	Knowledge: • Understand the concept of JDBC and the different types of JDBC drivers. • Familiarize with the JDBC packages and the overall process of establishing a database connection using JDBC. • Grasp the usage of Statement, Prepared Statement, and Callable Statement objects to execute SQL queries. • Recognize the importance of transaction processing, metadata retrieval, handling different data types, and managing exceptions in JDBC. Skills: • Ability to establish and manage database connections using JDBC in a Java application. • Proficient in executing SQL queries, processing Result Set data, and handling transactions within a JDBC context. • Competent in retrieving and interpreting metadata, managing data types, and handling exceptions in JDBC code. Attitudes: • Develop a meticulous approach to database interaction, ensuring efficient and secure data access through JDBC. • Foster an awareness of the importance of transaction management in maintaining data integrity and consistency. • Embrace best practices for error handling and resource management in JDBC to prevent issues such as resource leaks or SQL injection vulnerabilities.

3. Syllabus

ADVANCED JAVA **SEMESTER – V**			
Course Code	**23BCS505D**	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:

1.Understanding the fundamental concepts of Enumerations and Annotations
2.Apply the concepts of Generic classes in Java programs
3.Demonstrate the fundamental concepts of String operations
4.Design and develop web applications using Java servlets and JSP
5.Apply database interaction through Java database Connectivity

Module -1

Enumerations, Autoboxing and Annotations:

Enumerations: Enumeration fundamentals, the values() and valueOf() methods, Java enumerations are class types, enumerations inherits Enum, example, type wrappers,

Autoboxing,: Autoboxing methods, Autoboxing / Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors,

A word of warning Annotations, Annotation basics, specifying retention policy, obtaining annotations at run time by use of reflection, Annotated element interface, Using default values, Marker Annotations, Single member annotations, Built in annotations.

Textbook 1: Chapter 12

Module -2

String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueOf(), Changing the case of characters within a String, String Buffer, String Builder

Textbook 1: Chapter 15

Module -3

Applets: Introduction, Types of Applets, Applet Basics, Applet Architecture, An Applet Skeleton, Applet Initialization & Termination, Simple Applet Display Methods, HTML Applet Tag, Passing parameters & Applets. Applet Context and showdocument().
Swings: Introduction. Origins, Features, The MVC Connection, Componenets and Containers, Swing Packages, Simple Swing Application, Creating Swing Applet, Exploring Swings
Textbook 1: Chapter 23, 31, 32

Module -4

Servelet: The life cycle of a servlet; A simple servlet; the servlet API; The javax.servlet package, Reading servlet parameter; the javax.servlet.http package; Handling HTTP Requests and Responses; using Cookies; Session Tracking, Java Server Pages (JSP); JSP tags, Variables and Objects, Methods, Control statements, Loops, Request String, Parsing other information, User sessions, Cookies, Session Objects
Textbook 1: Chapter 38, Textbook 2: Chapter 11

Module -5

The concept of JDBC; JDBC Driver Types; JDBC packages; A brief overview of the JDBC Process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data Types; Exceptions.
Textbook 2: Chapter 6

TEXTBOOKS:
1. Herbert Schildt: JAVA the Complete Reference. 9th Edition, Tata McGraw-Hill
2. Jim Keogh, The Complete Reference J2EE, Tata McGraw-Hill

REFERENCE BOOKS:
1. Y. Daniel Liang: Introduction to JAVA Programming, 7th Edition, Pearson Education, 2007
2. Holzner,Steven, Java2 Programming Black Book, McGraw-Hill Education.

VIDEO LINKS:
1. https://nptel.ac.in/courses/106/105/106105191/
2. https://nptel.ac.in/courses/106/105/106105225/

4.Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3:	**Week 1: Enumerations** • **Topics:** ○ Enumeration fundamentals ○ values() and valueOf() methods ○ Java enumerations as class types ○ Enumerations inheriting Enum ○ Example of using enumerations • **Exercises:** ○ Practice creating enumerations and using methods like values() and valueOf(). ○ Implement a simple application using enums. **Week 2: Autoboxing** • **Topics:** ○ Introduction to Autoboxing/Unboxing ○ Autoboxing in expressions ○ Boolean and character values ○ Preventing errors with Autoboxing/Unboxing • **Exercises:** ○ Create examples showing how Autoboxing/Unboxing works. ○ Discuss scenarios where Autoboxing can prevent errors. **Week 3: Annotations** • **Topics:** ○ Annotation basics ○ Specifying retention policies ○ Obtaining annotations at runtime using reflection ○ AnnotatedElement interface

		○ Using default values, Marker Annotations, and Single-member annotations ○ Built-in annotations • **Exercises:** ○ Create custom annotations and retrieve them using reflection. ○ Discuss the use cases for marker annotations.
2	Week 4-6:	**Week 4: Generics: Introduction** • **Topics:** ○ What are Generics? ○ Simple Generics Example ○ Generic Class with Two Type Parameters ○ General Form of a Generic Class • **Exercises:** ○ Implement basic generic classes. ○ Discuss the importance of type safety with Generics. **Week 5: Generics: Advanced** • **Topics:** ○ Bounded Types and Using Wildcard Arguments ○ Bounded Wildcards ○ Creating a Generic Method ○ Generic Interfaces • **Exercises:** ○ Implement methods with bounded types and wildcards. ○ Practice creating generic interfaces. **Week 6: Generics: Expert Topics** • **Topics:** ○ Raw types and Legacy code ○ Generic Class Hierarchies ○ Erasure and Ambiguity errors ○ Some Generic Restrictions • **Exercises:** ○ Explore the concept of erasure and how it impacts code. ○ Analyze ambiguity errors in code and learn how to resolve them.
3	Week 7-8:	**Week 7: String Handling** • **Topics:** ○ The String Constructors, String Length ○ Special String Operations, Character Extraction ○ String Comparison, Searching Strings, Modifying a String ○ Data Conversion Using valueOf() ○ Changing the case of characters within a String • **Exercises:** ○ Practice with String manipulation methods. ○ Write programs that utilize the valueOf() method and string modification techniques. **Week 8: String Handling Continued** • **Topics:** ○ StringBuffer, StringBuilder • **Exercises:** ○ Compare the performance of String, StringBuffer, and StringBuilder in various scenarios. ○ Implement string manipulations using StringBuilder for efficiency.
4	Week 9:	**Week 9: Servlets** • **Topics:** ○ The life cycle of a servlet; A simple servlet; the servlet API ○ The javax.servlet package, Reading servlet parameters

		○ The javax.servlet.http package ○ Handling HTTP Requests and Responses; using Cookies; Session Tracking ○ Java Server Pages (JSP); JSP tags, Variables and Objects, Methods, Control statements, Loops • **Exercises:** ○ Develop a simple servlet-based application. ○ Implement session tracking and cookie handling.
5	Week 10:	**Week 10: JDBC** • **Topics:** ○ Concept of JDBC; JDBC Driver Types; JDBC packages ○ A brief overview of the JDBC Process ○ Database Connection ○ Associating the JDBC/ODBC Bridge with the Database ○ Statement Objects; ResultSet; Transaction Processing ○ Metadata, Data Types, Exceptions • **Exercises:** ○ Connect to a database using JDBC. ○ Execute SQL queries and handle results using JDBC.
10	Week 11-12	**Review and Project** • **Topics:** ○ Review of all topics ○ Discussion of key concepts and difficult areas • **Project:** ○ Develop a final project that integrates multiple topics from the syllabus (e.g., a web application using Servlets, JSP, JDBC with Generics and Annotations).

5.Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Problem-Based Learning (PBL)**	Engage students with real-world problems that require advanced Java knowledge to solve. For instance, tasks could involve optimizing a legacy application or implementing a micro services architecture. It Encourages deep understanding, critical thinking, and application of complex Java concepts in practical situations.
2	**Hands-On Coding Sessions**	Incorporate frequent coding exercises where students write, debug, and optimize Java code. Projects could include building scalable web applications, implementing multi-threaded programs, or designing custom data structures. It will enhances coding proficiency and reinforces theoretical knowledge through practical application
3	**Use of Advanced Java Frameworks and Libraries**	Introduce and work extensively with popular Java frameworks such as Spring, Hibernate, and Apache Kafka. Teach students how to integrate these tools into their projects. This will helps the students to prepare for industry demands, as these frameworks are widely used in enterprise-level development.
4	**Flipped Classroom Model**	Assign reading or video lectures as homework, and use class time for discussions, problem-solving sessions, and hands-on activities.
5	**Project-Based Learning**	Assign a capstone project where students must build a complete application from scratch, incorporating advanced Java concepts.
6	**Regular Assessments and Feedback**	Conduct quizzes, coding challenges, and peer assessments to regularly gauge student understanding. Provide detailed feedback to guide improvement.
7	**Guest Lectures and Industry Interaction**	Invite industry professionals to give talks on current trends, challenges, and opportunities in Java development. Arrange for students to work on live projects or case studies from the industry.

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

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	**Enumerations**	Understand the fundamentals of Java enumerations, including how to use values() and valueOf() methods, and explore their class-like properties.
2	**Autoboxing and Unboxing**	Grasp the concepts of autoboxing and unboxing in Java, including how these processes work with expressions and how they help prevent errors.
3	**Annotations**	Learn about Java annotations, their basics, and how to specify retention policies and obtain annotations at runtime.
4	**Generics**	Understand the use of generics in Java, including creating generic classes, methods, and understanding the restrictions and errors associated with generics.
5	**String Handling**	Master string manipulation and handling in Java, including various string operations and conversions.
6	**Servlets**	Understand the life cycle and working of Java servlets, including handling HTTP requests and responses.
7	**JDBC**	Learn the fundamentals of JDBC, including connecting to databases, executing queries, and handling database results.

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

Course Outcomes (COs)

Cos	Description
M23BCS505D.1	Understanding the fundamental concepts of Enumerations and Annotations
M23BCS505D.2	Apply the concepts of Generic classes in Java programs
M23BCS505D.3	Demonstrate the concepts of String operations in Java
M23BCS505D.4	Develop web based applications using Java servlets and JSP
M23BCS505D.5	Illustrate database interaction and transaction processing in Java

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS505D.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS505D.2	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCS505D.3	-	-	3	3	-	-	-	-	-	-	-	-	3	-
M23BCS505D.4	-	-	-	3	-	-	-	-	-	-	-	-	-	3
M23BCS505D.5	**-**	**-**	3	-	-	-	-	-	-	-	-	-	3	3

M23BCS505D	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

10.Future with this Subject:

1. **Career Opportunities**

- **Enterprise Application Development**: Advanced Java is extensively used in building enterprise-level applications, especially with frameworks like Spring, Hibernate, and Java EE.
- **Backend Development**: A solid grasp of Advanced Java can lead to roles focusing on backend development, where you manage server-side logic, databases, and integration with front-end components.
- **Mobile App Development (Android)**: While Android development has shifted towards Kotlin, a deep knowledge of Java is still valuable for maintaining and upgrading older applications.
- **Big Data Technologies**: Java is a preferred language for working with big data technologies like Apache Hadoop and Apache Kafka.
- **Cloud Computing**: With cloud services like AWS and Google Cloud supporting Java, there are numerous opportunities in cloud-based application development.
- **Microservices Architecture**: Advanced Java skills are critical in building and deploying microservices, which is a growing trend in software architecture.
- **DevOps Roles**: Java knowledge is essential in automating processes, continuous integration, and deployment pipelines.

2. Technological Advancements

- **Java in AI and ML**: Though Python is more popular for AI and ML, Java's ecosystem is growing, with libraries like Deeplearning4j making it relevant for these technologies.
- **Internet of Things (IoT)**: Java's portability makes it a strong contender for IoT applications, especially in embedded systems.
- **Blockchain Development**: Java's robustness and security features make it suitable for developing blockchain solutions.

3. High Demand and Job Security

- Companies consistently seek Java developers, ensuring long-term demand and job security.
- Many legacy systems still run on Java, so there’s always a need for skilled professionals to maintain and upgrade these systems.

4. Continuous Learning and Growth

- Mastering Advanced Java often leads to learning more about system architecture, design patterns, and best practices in software development.
- It also opens doors to learning and integrating other technologies like Python, JavaScript, and various cloud platforms, further enhancing your skill set.

5. Entrepreneurship

- With Advanced Java skills, you could start your own tech venture, offering software solutions, consulting, or creating products that serve a niche market.

5th Semester	**Project Work (PW)** **MINI PROJECT**	**M23BCS506**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Engineering Principles**	Fundamental courses in the respective engineering stream
2	**Application of Theoretical Knowledge in Practical Scenarios**	Knowledge of the core subjects of the respective stream
3	**Project Design and Planning**	Familiarity with design tools and project management techniques.
4	**Multidisciplinary Collaboration**	Basic knowledge of related disciplines (e.g., Mechanical students should have a basic understanding of Electronics, etc.).
5	**Technical Communication**	Writing technical reports and presenting technical content

2. Competencies

S/L	Competency	KSA Description
1	**Problem Identification and Analysis**	**Knowledge:** Understanding the problem domain and relevant engineering concepts. **Skill:** Ability to analyze and break down complex problems into manageable parts. **Attitude:** Attention to detail and a systematic approach to problem-solving.
2	**Solution Design and Implementation**	**Knowledge:** Familiarity with design methodologies and tools. **Skill:** Proficiency in creating prototypes or models using appropriate technologies. **Attitude:** Creativity and innovation in developing solutions.
3	**Interdisciplinary Collaboration**	**Knowledge:** Understanding of basic concepts from other engineering disciplines. **Skill:** Effective communication and teamwork in a multidisciplinary environment. **Attitude:** Openness to different perspectives and willingness to collaborate.
4	**Technical Documentation and Presentation**	**Knowledge:** Standards and practices for technical writing and reporting. **Skill:** Ability to document the project effectively and present it to an audience. **Attitude:** Confidence and clarity in communication.
5	**Project Management**	**Knowledge:** Understanding of project timelines, resource allocation, and risk management. **Skill:** Ability to plan, execute, and monitor a project from start to finish. **Attitude:** Responsibility and accountability in managing project tasks.

3. Project Timeline

S/L	Timeline	Description
1	**Week 1-2:** Introduction and Problem Definition	Students will define their project problem, scope, and objectives with the guidance of their mentors.
2	**Week 3-4:** Research and Feasibility Study	Conduct background research, explore existing solutions, and evaluate the feasibility of different approaches.
3	**Week 5-6:** Design and Planning	Develop a detailed project plan, including design specifications, timelines, and resource requirements.
4	**Week 7-8:** Prototype Development	Begin building the initial prototype or model, focusing on core functionalities.
5	**Week 9-10:** Testing and Refinement	Test the prototype, identify issues, and refine the design to improve performance.

6	**Week 11:** Final Implementation and Documentation	Complete the final implementation of the project and prepare detailed documentation.
7	**Week 12:** Presentation and Evaluation	Present the project to a committee for evaluation, followed by a Q&A session.

4. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation:

The CIE marks shall be awarded by a committee consisting of the Head of the concerned Department and two senior faculty members of the Department, one of whom shall be the Guide.

The CIE marks awarded for the project work, shall be based on the evaluation of the project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates.

SEE: There shall be no SEE.

5. Learning Objectives

S/L	Learning Objectives	Description
1	Identify and Analyze Engineering Problems	Students will learn to identify real-world engineering problems, analyze them, and propose feasible solutions.
2	Design and Implement Solution(s)	Students will gain experience in designing and implementing engineering solutions using appropriate tools and methodologies.
3	Collaborate Effectively in Teams	Students will develop teamwork skills through collaboration with peers from different engineering disciplines.
4	Communicate Technical Information	Students will enhance their ability to document and present technical information effectively.

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

Course Outcomes (COs)

Cos	Description
M23BCS506.1	Apply engineering principles to identify, formulate, and solve real-world problems.
M23BCS506.2	Design and develop prototypes or models that address specific engineering challenges.
M23BCS506.3	Collaborate with team members to complete the project successfully.
M23BCS506.4	Document and present the project effectively, demonstrating clear communication skills.

CO-PO-PSO Mapping

COs/Pos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS506.1	3	3							3			3	3	
M23BCS506.2	3	3	3				3				3	3	3	
M23BCS506.3		3	2	3		3						3	3	
M23BCS506.4				3	3			3		3		3	3	
M23BCS506	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	

6. Future with this Subject

The mini-project course will serve as a foundation for more complex and comprehensive project work in the final year, such as the capstone project. The skills developed here, including problem-solving, design, teamwork, and communication, will be crucial for successful completion of future courses and for professional practice in engineering.

5th Semester	**ABILITY ENHANCEMENT(AE)** **Research Methodology and IPR**	**M23BRMK507**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Understanding of Research Concepts:	Students should have a foundational understanding of what research entails, including its purpose, types, and significance in academic and professional contexts. **Research and Its Types**: Know the differences between basic and applied research, as well as qualitative, quantitative, mixed methods, Variable, Understand dependent, independent, and control variables. **Hypothesis Formation:** Grasp how to formulate and test hypotheses.
2.	Familiarity with Research Design	**Technical Writing:** Proficiency in academic and technical writing is essential. This includes the ability to write clear, concise, and well-structured research papers, essays, and reports. **Experimental Designs:** Learn about different experimental setups, such as controlled experiments, field experiments, and natural experiments. **Non-Experimental Designs**: Understand case studies, longitudinal studies, and cross-sectional studies.
3.	Statistical and Analytical Skills:	**Descriptive Statistics**: Know how to summarize and describe data using measures of central tendency and variability. **Inferential Statistics:** Understand concepts like hypothesis testing, confidence intervals, and p-values. **Data Analysis Software:** Familiarity with tools like SPSS, R, or Python for analyzing data.
4.	Familiarity with Intellectual Property Concepts:	**Introduction to IPR**: A basic understanding of intellectual property rights, including what they are, the different types (patents, trademarks, copyrights, trade secrets), and their significance. **IP Law Basics:** Awareness of basic IP laws and regulations, both at the national and international levels, as they relate to protecting intellectual property. **Ethical Considerations**: Understanding the ethical considerations in research and IP, including issues like copyright infringement, patent rights, and the moral obligations of researchers.
5.	Background in Relevant Academic Discipline:	**Subject-Specific Knowledge**: Depending on the focus of the research methodology course, students should have foundational knowledge in the specific academic discipline (e.g., engineering, social sciences, business) to which the research methodologies will be applied. **Problem-Solving Skills**: Strong problem-solving skills to apply research methodologies effectively within their field of study.
6	Communication and Collaboration Skills:	**Effective Communication**: Ability to effectively communicate research ideas, methodologies, and findings both in written and verbal formats. **Collaboration:** Experience working in teams, as research often involves collaborative efforts, requiring the ability to work effectively with others.
7	Computer and Internet Proficiency:	**Research Tools:** Familiarity with online databases, academic journals, and research tools like Google Scholar, JSTOR, and others for conducting literature reviews and gathering research material. **Document Preparation**: Proficiency in using word processors (e.g., Microsoft Word) and presentation tools (e.g., PowerPoint) to prepare and present research findings. These prerequisites ensure that students are adequately prepared to engage with the material covered in Research Methodology and IPR courses, enabling them to conduct meaningful research and understand the complexities of intellectual property rights.
8	Critical Thinking and Problem-Solving:	**Analytical Skills**: Develop the ability to critically assess research methods and results. **Problem-Solving:** Be able to identify potential issues in research design and implementation.

2. Competencies

S/L	Competency	KSA Description
1	Research Methodology and Intellectual Property Rights (IPR)	**Knowledge:** Research Design, Data Collection Methods, Statistical Analysis, Research Ethics, And Intellectual Property Rights (IPR): Types of Intellectual Property, IPR Laws and Regulations, IP Management. **Skills:** Critical Analysis, Data Management, Technical Writing, Project Management, And Intellectual Property Rights (IPR): Legal Research, IP Documentation, Negotiation and Licensing, Ethical Decision-Making. **Attitudes:** Curiosity and Open-Mindedness, Integrity and Ethical Responsibility, Persistence and Resilience, Intellectual Property Rights: Respect for Intellectual Property, Proactive Protection, Collaboration and Fairness.
2	Defining the Research Problem	**Knowledge:** Concept of a Research Problem, Importance of the Research Problem, Contextual Knowledge, Types of Research Problems **Skills:** Critical Thinking, Literature Review, Clarity and Precision. Problem Formulation: Narrowing the Focus, Feasibility Assessment, Hypothesis Development. Alignment with Objectives: Alignment with Research Goals, Stakeholder Consideration. **Attitudes:** Curiosity and Inquisitiveness: Desire to Explore, Open-Mindedness, Critical Reflection: Reflective Thinking, Adaptability, Ethical Responsibility: Ethical Consideration, Responsibility to the Field.
3	Reviewing the literature	**Knowledge:** Understanding of Literature Review: Purpose of a Literature Review, Types of Literature, Theoretical Frameworks, Literature Search Strategies: Search Tools and Databases, Keywords and Boolean Operators, Critical Reading and Analysis: Evaluating Sources, Synthesizing Information. **Skills:** Literature Search: Efficient Searching, Citation Management, Critical Evaluation: Analytical Reading, Comparative Analysis, Synthesis and Writing: Integrating Literature, Structured Writing, Paraphrasing and Quoting, Ethical Use of Literature: Avoiding Plagiarism, Bias Identification. **Attitudes:** Curiosity and Open-Mindedness: Intellectual Curiosity, Open-Mindedness, Critical Reflection: Skeptical Inquiry, Reflective Thinking, Respect for Academic Integrity: Ethical Responsibility, Responsibility to the Field.
4	Research Design	**Knowledge:** Research design refers to the overall strategy utilized to answer research questions. A research design typically outlines the theories and models underlying a project. **Skills:** Ability to search for, locate, extract, organize, evaluate and use or present information that is relevant to a particular topic **Attitudes:** A process of detailed and methodical investigation into some area of study.
5	Design of Sample Surveys	**Knowledge:** Sample survey design is a very mature and deeply rooted discipline in the statistical literature. **Skills:** Mastering the art (and science) of running your own surveys takes time and practice. **Attitudes**: A study design is a set of decisions (design parameters) about what and how much data to collect, and when (how often, for how long) and where to collect it.
6	Testing of Hypotheses	**Knowledge:** Understand statistical method, used to determine if there is enough evidence in a sample data to draw conclusions about a problem statement. **Skills:** Hypothesis testing is a fundamental concept in statistics that allows us to draw conclusions about a population based on a sample of data. It is a systematic approach used to evaluate whether a claim or hypothesis about the population is supported by the evidence provided by the sample. **Attitudes:** Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data. The test provides evidence concerning the plausibility of the hypothesis, given the data.
7	Intellectual Property:	**Knowledge:** Protects the work of inventors, artists, writers, and creators, as well as the reputation of individuals, brands, and organization.

		Skills: Refers to creations of the mind, such as inventions, literary ,artistic works and designs; **Attitudes:** IPR provide certain exclusive rights to the inventors or creators of that property, in order to enable them to reap commercial benefits from their creative efforts or reputation.

3. Syllabus

Research Methodology and IPR
SEMESTER – V

Course Code	**M23BRMK507**	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:

1. To give an overview of the research methodology and explain the technique of defining a research problem.
2. To explain the functions of the literature review in research..
3. To explain carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.
4. To explain various research designs and their characteristics
5. To explain the details of sampling designs, measurement and scaling techniques and also different methods of data collections
6. To explain the art of interpretation and the art of writing research reports.
7. To explain various forms of the intellectual property, its relevance and business impact in the changing global business environment.

Module -1

Research Methodology: Introduction, Meaning of Research, Objectives of Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Research Process, Criteria of Good Research.

Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem.

Module -2

Defining the Research Problem: Technic involved in the defining a problem.

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic principles of experimental design,

Module -3

Design of Sample Surveys: Design of Sampling: Introduction, Sample Design, Sampling and Non-sampling Errors, Sample Survey versus Census Survey, Types of Sampling Designs**.**

Measurement and Scaling: Qualitative and Quantitative Data, Classifications of Measurement Scales, Goodness of Measurement Scales, Sources of Error in Measurement, Techniques of Developing Measurement Tools, Scaling, Scale Classification Bases, Scaling Technics.

Module -4

Testing of Hypotheses: Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Testing of Hypothesis, Test Statistics and Critical Region, Critical Value and Decision Rule, Procedure for Hypothesis Testing, Hypothesis Testing for Mean, Proportion, Variance, for Difference of Two Mean, for Difference of Two Proportions, for Difference of Two Variances, P-Value approach, Power of Test, Limitations of the Tests of Hypothesis**.**

Module -5

Interpretation and Report Writing: Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout of the Research Report.

Intellectual Property: The Concept, Intellectual Property System in India, Development of TRIPS Complied Regime in India, Patents Act, 1970, Trade Mark Act, 1999,The Designs Act, 2000, The Geographical Indications of Goods (Registration and Protection) Act1999, Copyright Act,1957,The Protection of Plant Varieties and Farmers' Rights Act, 2001,The Semi-Conductor Integrated Circuits Layout Design Act, 2000, Trade Secrets, Utility Models, IPR and Biodiversity, The Convention on Biological Diversity (CBD) 1992, Competing Rationales for Protection of IPRs, Leading International Instruments Concerning IPR, World Intellectual Property Organization (WIPO).

TEXTBOOKS:

1. Research Methodology: Methods and Techniques C.R. Kothari, Gaurav Garg New Age International 4th Edition, 2018.

2.Study Material (For the topic Intellectual Property under module 5) Professional Program Intellectual Property Rights, Law and Practice, The Institute of Company Secretaries of India, Statutory Body Under an Act of Parliament, September 2013

Reference Books

1. Research Methods: the concise knowledge base Trochim Atomic Dog Publishing 2005
2. Conducting Research Literature Reviews: From the Internet to Paper Fink A Sage Publications 2009

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: Research Methodology and Defining the Research Problem	**Competency:** Understand the research methodology and problem statements on research. If necessary defines the problem. **Knowledge:** Understand the various functions Research Methods. **Skills**: Know the functions of Research Methods versus Methodology, Research and Scientific Method, Research Process, Criterias of Good Research.
2	Week 4-6: Reviewing the literature and Research Design	**Competency:** Understand the concept of reviewing the literature and broadening knowledge base in research area. **Knowledge**: Understanding of Literature Review: Purpose of a Literature Review, Types of Literature, Theoretical Frameworks, Literature Search Strategies: Search Tools and Databases, Keywords and Boolean Operators, Critical Reading and Analysis. **Skills**: and able to identify Features of a Good Design, Important Concepts Relating to Research Design.
3	Week 8-11: Design of Sample Surveys: Design of Sampling and Measurement and Scaling:	**Competency:** Studying the different types, Sampling and Non-sampling Errors, Sample Survey versus Census Survey. **Knowledge:** Sample survey design is a very mature and deeply rooted discipline in the statistical literature. **Skills:** Learn the qualitative and Quantitative Data, Classifications of Measurement Scales, Goodness of Measurement Scales, and Sources of Error in Measurement.
4	Week 7-8: Testing of Hypotheses	**Competency:** Studying the concept of Basic Concepts Concerning Testing of Hypotheses, Testing of Hypothesis and Test Statistics. **Knowledge**: Understand statistical method, used to determine if there is enough evidence in a sample data to draw conclusions about a problem statement. **Skills:** Able to learn the critical Region, Critical Value and Decision Rule, Procedure for Hypothesis Testing and limitation.
5	Week 9-12: Interpretation and Report Writing and Intellectual Property:	**Competency:** Studying Types of Intellectual Property, IPR Laws and Regulations, IP Management. **Knowledge**: Research Design, Data Collection Methods, Statistical Analysis, Research Ethics, and Intellectual Property Rights (IPR): **Skills**: Learning report writing and The Concept, Intellectual Property System in India, Development of TRIPS Complied Regime in India, Patents Act, 1970, Trade Mark Act, 1999,The Designs Act, 2000, The Geographical Indications of Goods (Registration and Protection) Act1999, Copyright Act,1957

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Lectures and Seminars	**Engaging Presentations:** Use interactive lectures to introduce key concepts in research methodology and IPR. Incorporate multimedia presentations, real-world examples, and case studies to make complex topics accessible and engaging. **Guest Lectures**: Invite experts in research methodology, data analysis, and IPR to provide insights from their professional experiences. This helps bridge the gap between theory and practice and offers students diverse perspectives.
2	Case-Based Learning	**Real-Life Scenarios:** Utilize case studies that illustrate the application of research methodology and IPR in various fields. For example, examine how companies protect their intellectual property, or how researchers design studies to answer specific research questions.

		Problem-Solving Exercises: Present students with real or hypothetical research problems or IP issues and have them work in groups to develop solutions. This fosters critical thinking and collaborative skills.
3	Research Projects and Assignments	**Practical Research Assignments**: Assign students individual or group research projects where they must define a research problem, review literature, design a study, and analyze data. This hands-on approach reinforces theoretical concepts. **IPR Analysis Projects**: Students can be tasked with evaluating the IP strategy of a particular company or analyzing a legal case involving intellectual property. This encourages the application of IPR knowledge to real-world situations.
4	Workshops and Hands-On Training	**Data Analysis Workshops**: Conduct workshops that provide students with practical training in data analysis software (e.g., SPSS, R, NVivo). These sessions should include step-by-step guidance on how to process and analyze research data. **IP Documentation Practice:** Organize workshops where students learn to draft and file patents, trademarks, and copyrights. This practical experience is crucial for understanding the technical and legal aspects of IP management.
5.	Collaborative Learning	**Group Discussions:** Facilitate small group discussions where students can debate research methodologies or the ethical implications of IP laws. This encourages peer-to-peer learning and helps students articulate their understanding. **Peer Review:** Implement peer review processes where students evaluate each other's research proposals or IP strategies. This not only improves their critical thinking but also fosters a collaborative learning environment.

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
	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 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 Research Fundamentals	To give an overview of the research methodology and explain the technique of defining a research problem. To explain the functions of the literature review in research..

2	Conducting Literature Reviews	To explain carrying out a literature search, its review, developing theoretical and conceptual frameworks and writing a review.
3	Designing Research Methodologies	To analyse the various research designs and their characteristics
4	Data Collection and Analysis	To analyse and understand the details of sampling designs, measurement and scaling techniques and also different methods of data collections
5	Understanding Intellectual Property Rights (IPR)	To explain the art of interpretation and the art of writing research reports and Analyse various forms of the intellectual property, its relevance and business impact in the changing global business environment

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

Course Outcomes (COs)

Cos	Description
M23BRMK507.1	Understand the research methodology and problem along with scientific method and process.
M23BRMK507.2	Analyse the Literature Review and research design process.
M23BRMK507.3	Design the sample survey and measurement and scaling.
M23BRMK507.4	Analyse the testing hypothesis of variance and limitations of research concepts.
M23BRMK507.5	Apply and analyse the research report and The Geographical Indications of Goods (Registration and Protection) Act1999,World Intellectual Property Organisation

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BRMK507.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BRMK507.2	-	3	3	-	-	-	-	-	-	-	-	-	3	-
M23BRMK507.3	-	3	3	-	-	-	-	-	-	-	-	-	3	-
M23BRMK507.4	-	3	3	-	-	-	-	-	-	-	-	-	3	-
M23BRMK507.5	-	-	3	-	-	-	-	-	-	-	-	-	3	-
M23BRMK507	**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

10. Future with this Subject:

1. Academic Advancement

- **Pursuing Higher Education:** Mastery of research methodology and IPR provides a strong foundation for advanced studies, such as pursuing a Ph.D. or other research-intensive postgraduate programs. This subject equips students with the necessary skills to conduct independent research and contribute original knowledge to their field.

- **Academic Publishing:** Understanding research methodology enhances the ability to publish research findings in peer-reviewed journals, contributing to academic discourse and establishing a scholarly reputation.

2. Professional Opportunities

- **Research and Development (R&D):** Proficiency in research methodology is critical for careers in R&D across various industries, including technology, pharmaceuticals, and social sciences. Professionals can lead or contribute to innovation projects, product development, and experimental research.
- **Intellectual Property Management:** Knowledge of IPR opens up career paths in IP management, including roles such as patent agents, IP consultants, or legal advisors in technology transfer offices or law firms specializing in IP rights.
- **Consulting and Advisory Roles:** Expertise in research methodology and IPR can lead to consulting opportunities, where professionals provide advice on research design, data analysis, and IP strategy to businesses, government agencies, or non-profit organizations.

3. Entrepreneurship and Innovation

- **Startups and Innovation:** Understanding IPR is crucial for entrepreneurs and innovators who wish to protect their inventions, trademarks, or creative works. This knowledge enables them to navigate the legal landscape, secure patents, and manage IP assets effectively.
- **Technology Transfer:** Professionals with expertise in research methodology and IPR can work in technology transfer, facilitating the commercialization of research innovations from universities and research institutions to the market.

4. Contribution to Society

- **Policy Development:** Expertise in research methodology and IPR is valuable for contributing to policy development in areas such as innovation, education, healthcare, and intellectual property law. Professionals can work with governmental and international organizations to shape policies that promote ethical research and protect intellectual property rights.
- **Social Impact Research:** Professionals can engage in research that addresses societal challenges, such as public health, environmental sustainability, or social equity. Understanding research methodology ensures that such research is rigorous, valid, and capable of informing public policy and practice.

5. Lifelong Learning and Professional Development

- **Continuous Education:** The field of research methodology and IPR is dynamic, with ongoing advancements in research techniques, data analysis tools, and IP laws. Professionals committed to this subject will engage in lifelong learning, attending workshops, conferences, and courses to stay current with developments.
- **Networking and Collaboration:** Future opportunities in this subject involve building strong professional networks and collaborating with experts across disciplines. This collaborative approach enhances research quality and broadens the impact of one's work.

6. Ethical and Sustainable Research

Promoting Ethical Research: As research and innovation continue to evolve, there will be an increasing focus on ethical considerations, including the responsible conduct of research, equitable access to IP, and the sustainable use of resources. Professionals in this field will play a critical role in promoting ethical research practices and advocating for policies that balance innovation with social responsibility.

7. Influence on Future Innovations

Shaping the Future of Research and IP: Mastery of research methodology and IPR enables professionals to influence the direction of future research and innovation. Whether through groundbreaking research, the development of new IP strategies, or contributions to the legal framework, individuals in this field will be at the forefront of shaping how knowledge is created, shared, and protected.

5thSemester	**Basic Science (BS)** **ENVIRONMENTAL STUDIES**	**M23BESK508**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Understanding Ecosystems**	Basic knowledge of biology, environmental science, and ecological systems.
2	**Comprehending Natural Resource Management**	Familiarity with energy systems, environmental management, and global sustainability practices.
3	**Knowledge of Environmental Pollution**	Understanding of chemical processes, industrial impacts, and environmental science fundamentals.
4	**Addressing Global Environmental Concerns**	Knowledge of climate science, environmental policies, and global ecological challenges.
5	**Awareness of Environmental Legislation**	Familiarity with national and international environmental laws, policies, and regulations.

2. Competencies

S/L	Competency	KSA Description
1	**Ecosystem Analysis**	**Knowledge**: Ecosystem structure, sustainability principles, SDGs. **Skills**: Identifying ecosystem components, and understanding sustainability targets. **Attitudes**: Appreciating biodiversity, and promoting sustainability.
2	**Resource Management**	**Knowledge**: Renewable and non-renewable energy systems, sustainable practices. **Skills**: Analyzing case studies, and evaluating energy systems. **Attitudes**: Supporting sustainable resource use, and critical thinking on global issues.
3	**Pollution Mitigation**	**Knowledge**: Pollution sources, impacts, and legislation. **Skills**: Assessing pollution control measures, and implementing waste management strategies. **Attitudes**: Advocating for environmental protection, and responsible waste disposal.
4	**Global Environmental Awareness**	**Knowledge**: Climate change, groundwater depletion, global policies. **Skills**: Investigating global environmental challenges, and proposing solutions. **Attitudes**: Engaging in global environmental discussions, and supporting international efforts.
5	**Environmental Legal Framework**	**Knowledge**: Key environmental acts and regulations. **Skills**: Applying legal knowledge to environmental issues, and understanding EIA processes. **Attitudes**: Valuing legal frameworks, and ensuring compliance with environmental laws.

3. Syllabus

ENVIRONMENTAL STUDIES
SEMESTER – V

Course Code	**M23BESK508**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:0:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**25 Hours**	Total Marks	**100**
Credits	**02**	Exam Hours	**02**

Course Objectives: Students will be able

1. Understand the structure and function of various ecosystems like forests, deserts, wetlands, rivers, oceans, and lakes.
2. Explore natural resource management techniques, including energy systems and disaster management, and assess their sustainability.
3. Examine environmental pollution sources and impacts, and learn corrective and preventive measures alongside waste management strategies.

4. Investigate global environmental issues such as climate change and groundwater depletion, and the role of environmental legislation in addressing these issues.

Module -1

ECOSYSTEMS (STRUCTURE AND FUNCTION): Forest, Desert, Wetlands, River, Oceanic and Lake. Sustainability: 17 SDGs- History, targets, implementation , Capacity Development

Module -2

NATURAL RESOURCE MANAGEMENT

Advances in Energy Systems (Merits, Demerits, Global Status and Applications): Hydrogen, Solar, OTEC, Tidal and Wind.

Natural Resource Management (Concept and case-studies): Disaster Management, Sustainable Mining - case studies and Carbon Trading.

Module -3

ENVIRONMENTAL POLLUTION & WASTE MANAGEMENT Environmental Pollution (Sources, Impacts, Corrective and Preventive measures, Relevant Environmental Acts, Case-studies): Surface and Ground Water Pollution; Noise pollution; Soil Pollution and Air Pollution. Waste Management: Bio-medical Wastes; Solid waste; Hazardous wastes; E-wastes; Industrial and Municipal Sludge.

Module -4

Global Environmental Concerns (Concept, policies and case-studies): Ground water depletion/recharging, Climate Change; Acid Rain; Ozone Depletion; Radon and Fluoride problem in drinking water; Resettlement and rehabilitation of people, Environmental Toxicology.

Module -5

ENVIRONMENTAL LEGISLATION : Water Act 1974, Air Act 1981, Environmental Protection Act 1984, Solid Waste Management Rules-2016, E- Waste management Rule - 2022, Biomedical Waste management-2016. Environmental Impact Assessment

TEXTBOOKS:

1. Environmental studies, Benny Joseph, Tata Mcgraw-Hill 2nd edition 2012 \
2. Environmental studies, S M Prakash, pristine publishing house, Mangalore 3rd edition-2018

REFERENCE BOOKS:

1. Benny Joseph, Environmental studies, Tata Mcgraw-Hill 2nd edition 2009
2. M.Ayi Reddy Textbook of environmental science and Technology, BS publications 2007
3. Dr. B.S Chauhan, Environmental studies, university of science press 1st edition

VIDEO LINKS:

1. Weblink:https://sdgs.un.org/goals Video Lectures
2. https://archive.nptel.ac.in/courses/109/105/109105190/ .

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1-2**	Introduction to ecosystems, exploring their structure and function with a focus on sustainability and SDGs.
2	**Week 3-4**	Understanding natural resource management, advances in energy systems, and disaster management through case studies.
3	**Week 5-6**	Examination of environmental pollution sources, impacts, and preventive measures, along with waste management strategies.
4	**Week 7-8**	Exploration of global environmental concerns such as climate change, groundwater depletion, and related policies.
5	**Week 9-10**	Study of environmental legislation, including key environmental acts and the process of Environmental Impact Assessment (EIA).
6	**Week 11-12**	Revision

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Lectures:	Utilize chalk and talk along with PowerPoint presentations and animations to engage students in theoretical and practical understanding
2	Case Study Analysis:	Present real-world scenarios and case studies to help students apply theoretical knowledge to practical situations, particularly in natural resource management and pollution control.

3	Fieldwork and Site Visits	Encourage hands-on learning through field visits to environmental labs, green buildings, and treatment plants, followed by documentation and analysis of the processes observed.
4	Collaborative Learning	Promote group projects and discussions, enabling students to collaborate and learn from each other, particularly in global environmental concerns and energy systems.

6. Assessment Details (both CIE and SEE)

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 mark. **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% of the maximum marks meant for SEE.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Analyze the structure and function of various ecosystems.	Students will learn about the characteristics and interactions within ecosystems such as forests, deserts, wetlands, rivers, oceans, and lakes.
2	Evaluate natural resource management techniques.	Students will assess the merits and demerits of various energy systems and learn sustainable management practices through case studies.
3	Investigate environmental pollution and waste management.	Students will understand the sources and impacts of environmental pollution, along with strategies for pollution control and waste management.
4	Explore global environmental concerns and policies.	Students will study global issues like climate change and groundwater depletion, and examine the role of environmental legislation in addressing these challenges.
5	Understand environmental	Students will gain insights into key environmental acts and regulations, and learn how to apply them in real-world scenarios.

	legislation and its application.	

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

Course Outcomes (COs)

Cos	Description
M23BESK508.1	Analyze the structure and functions of various ecosystems and evaluate their sustainability
M23BESK508.2	Apply knowledge of natural resource management and advances in energy systems to assess their global impacts
M23BESK508.3	Investigate environmental pollution sources and apply waste management strategies in real-world scenarios
M23BESK508.4	Critically analyze global environmental concerns and assess the effectiveness of environmental policies
M23BESK508.5	Demonstrate an understanding of environmental legislation and apply it to ensure sustainable practices

CO-PO-PSO Mapping

COs/POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
M23BESK508. 1	3	-	-	-	-	-	-	-	-	-	-	3
M23BESK508. 2	-	3	-	-	-	-	-	-	-	-	-	3
M23BESK508. 3	-	-	3	-	-	-	-	-	-	-	-	3
M23BESK508. 4	-	-	-	3	-	-	-	-	-	-	-	3
M23BESK508. 5	-	-	-		3	-	-	-	-	-	-	3
M23BESK508	**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**

10. Future with this Subject

This course provides a foundational understanding of environmental science that is crucial for advanced studies in environmental engineering, sustainability, and policy-making. It equips students with the knowledge and skills to tackle global environmental challenges and supports interdisciplinary research, making it a valuable asset for careers in environmental management, consulting, and advocacy. The insights gained from this course will also be beneficial in professional roles requiring compliance with environmental legislation and sustainable development practices.

5thSemester	**Non-Credit Mandatory Course(NCMC) National Service Scheme(NSS)**	**M23BNSK509**

Non-Credit Mandatory Courses(NCMC)
National Service Scheme(NSS)

Course Code	**M23BNSK509**		
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**
Activities Report Evaluation by College NSS Officer at the end of every semester(3rd to 6thsemester)			

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 themselves 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. In addition 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 homework, 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 stakeholders–Implementation.
5. Preparing an action able business proposal for enhancing the village in come and approach for implementation.
6. Helping local schools to achieve good results and enhance their enrolment in 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 connect and 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:

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.

At the end of every semester, activity report should be submitted for evaluation.

Distribution of Activities – Semester wise from 3rd to 6th semester.

Sem	Topics / Activities to be Covered
3rdSem for 25 Marks	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.
4thSem for 25 Marks	1. Water conservation techniques– Role of different stakeholders–Implementation. 2. Preparing an actionable 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.
5thSem 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. For eg. Digital India, Skill India, SwachhBharat, Atmanirbhar Bharath, MakeinIndia, Mudrascheme, Skill development programs etc. 3. Spreading public awareness under rural out reach programs.(minimum 5 programs). 4. Social connect and responsibilities.
6thSemfor 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 (Course Skill Set):

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

Cos	Description
M23BNSK509.1	Understand the importance of his/her responsibilities towards society.
M23BNSK509.2	Analyse the environmental and societal problems/issues and will be able to design solutions for the same.
M23BNSK509.3	Evaluate the existing system and to propose practical solutions for the same for sustainable development.
M23BNSK509.4	Implement government or self-driven projects effectively in the field.
M23BNSK509.5	Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

Pedagogy–Guidelines

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/ 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
2.	Waste management– Public, Private and Govt organization,5R's.	May be individual or team	Villages/ City Areas / Grama panchayat/ public	Site selection / proper consultation/ Continuous monitoring/ Information	Report should be submitted by an individual to the	Evaluation as per the rubrics of the scheme and

			associations/ Government Schemes officers /campus etc…	board	concerned evaluation authority	syllabus by NSS officer
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 / College 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
4.	Water conservation techniques – Role of different stakeholders– Implementation.	May be individual or team	Villages/ C ity 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
5.	Preparing an actionable business proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/ C ity 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
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 Schemesofficers /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
	Contribution to any			Group		

8.	national-level initiative of the Government of India. For eg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudrascheme, Skill development programs etc.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers /campus etc…	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 out reach programs.(minimum 5 programs).Social connect and responsibilities.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers /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 Schemesofficers /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

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

Plan of Action ((Execution of Activities For Each Semester)

Sl. No	**Practice Session Description**
1.	**Lecture session by NSS Officer**
2.	**Students Presentation on Topics**
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.	**Case-study-based Assessment, Individual performance**
11.	**Sector wise study and its consolidation**
12.	**Video-based seminar for 10-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.
- At last in 6th semester consolidated report of all activities from 3rd to 6thsemester, compiled report should be submitted as per the instructions.

Assessment Details:

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 the NSS Officer, the HOD, and the principal. • At-last Report should be 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.
Commencement 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 byeach student At the end 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 the field.

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

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.

5th Semester	**Non-Credit Mandatory Courses(NCMC)** **PHYSICAL EDUCATION (SPORTS & ATHLETICS) -III**	**M23BPEK509**

Non-Credit Mandatory Course (NCMC)
PHYSICAL EDUCATION (SPORTS & ATHLETICS) -III

Course Code	**M23BPEK509**	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	**-**

Semester-V

PHYSICAL EDUCATION (SPORTS & ATHLETICS)—III

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

COs	**Description**
M23BPEK509.1	Understand the ethics and moral values in sports and athletics.
M23BPEK509.2	Perform in the selected sports or athletics of the student's choice.
M23BPEK509.3	Understand the roles and responsibilities of organization and administration of sports and games.

Module-1

Ethics and Moral Values **(5hours)**

A. Ethics in Sports
B. Moral Values in Sports and Games

Module-2

Specific Games(Anyone to be selected by the student) **(20hours)**

A. Volley ball—Attack,Block,Service,Upper Hand Pass and Lower hand Pass.
B. Throwball—Service,Receive,Spinattack,NetDrop & Jumpthrow.
C. Kabaddi—Handtouch,ToeTouch,ThighHold,Anklehold and Bonus.
D. Kho-Kho—Giving Kho,SingleChain,Pole dive,Pole turning,3-6Up.
E. TableTennis—Service(ForeHand&BackHand),Receive(ForeHand&BackHand),Smash.
F. Athletics(Track/FieldEvents)—Any event as per availability of Ground.

Module-3

Role of Organisation and administration (5 hours)

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

5thSemester	**Non-Credit Mandatory Course(NCMC)** **Yoga**	**M23BYOK509**

Non-Credit Mandatory Courses(NCMC)
Yoga

Course Code	**M23BYOK509**		
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-Voce

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 tranquility
  3. Contentment

Yoga Syllabus

Semester V

- Patanjali's Ashtanga 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. Sheetali 3. Sheektari

Course outcomes (Course Skill Set):
At the end of the course, the student will be able to:

COs	**Description**
M23BYOK509.1	Understand the meaning, aim and objectives of Yoga.
M23BYOK509.2	Perform Suryanamaskar and able toTeach its benefits.
M23BYOK509.3	Understand and teach different Asanasbyname,its importance,methods and benefits.
M23BYOK509.4	Instruct Kapalabhati and its need and importance.
M23BYOK509.5	Teach different types of Pranayamaby its name,precautions,procedure and uses
M23BYOK509.6	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 (Twointernal 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

6. https://youtu.be/KB-TYlgd1wE

7. https://youtu.be/aa-TG0Wg1Ls

6th Semester	**Integrated Professional course (IPC)** **Full Stack Development**	**M23BCS601**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Programming Knowledge	Python: Since Django is a Python-based framework, strong proficiency in Python is essential. You should be comfortable with Python syntax, data types, loops, functions, and object-oriented programming (OOP).
2	Front-End Development Skills	HTML/CSS: Proficiency in HTML and CSS is necessary to create the structure and style of web pages. You should understand the basics of responsive design. JavaScript: Knowledge of JavaScript is important for adding interactivity to web pages. Understanding the Document Object Model (DOM) and how to manipulate it using JavaScript is crucial. Front-End Frameworks/Libraries: Familiarity with front-end frameworks like Bootstrap for styling, or JavaScript libraries like jQuery, can be beneficial. Learning a modern front-end framework like React can also be helpful for creating dynamic user interfaces.
3	Django-Specific Knowledge	Django Framework: Proficiency in Django is key. This includes understanding Django's MVC (Model-View-Controller) architecture, working with models, views, and templates, and using Django's ORM (Object-Relational Mapping) to interact with databases.
4	Database Knowledge	SQL Databases: Understanding of SQL and how to work with databases like PostgreSQL, MySQL, or SQLite, which are commonly used with Django. You should be comfortable with designing database schemas, writing queries, and optimizing database performance. Django ORM: Proficiency in Django's ORM to perform database operations in a Pythonic way without writing raw SQL.
5	Security Practices	Django Security Features: Familiarity with Django's built-in security features, such as CSRF protection, SQL injection protection, and handling authentication securely. Web Security Basics: Understanding of HTTPS, secure password storage, user authentication and authorization, and common web vulnerabilities.

2. Competencies

S/L	Competency	KSA Description
1	**Web Technologies**	**Knowledge:** Frontend: HTML5, CSS3, JavaScript, frameworks like React.js, Angular, or Vue.js. Backend: Server-side languages like Node.js, Java, or Python, and frameworks (Express.js, Django). **Skills:** Develop responsive web applications that deliver a seamless user experience. **Attitudes:** Build interactive and dynamic user interfaces as well as robust server-side applications.
2	**Database Management**	**Knowledge:**Familiarity with relational databases (MySQL, PostgreSQL) and NoSQL databases (MongoDB) **Skills:** Write complex queries and optimize database performance. **Attitudes:** Design, implement, and manage databases efficiently.
3	**API Development**	**Knowledge:** Knowledge of RESTful APIs and GraphQL. **Skills:** Test and document APIs for usability. **Attitudes:** Design and implement APIs that facilitate communication between frontend and backend.
4	**Responsive Web Design**	**Knowledge**: Familiarity with responsive design principles and frameworks (e.g., Bootstrap). **Skills**: Use CSS techniques to ensure design consistency across devices. **Attitudes**: Create interfaces that adapt to various screen sizes.

5	**Problem-Solving**	**Knowledge**: Techniques for troubleshooting and debugging. **Skills**: Implement solutions efficiently and evaluate their effectiveness **Attitudes:** Analyze complex issues and propose effective solutions.
6	**Team Collaboration**	**Knowledge:** Principles of teamwork and collaboration in software development. **Skills:** Communicate clearly and constructively in team settings. **Attitudes:** Work effectively in diverse teams and contribute to group objectives.
7	**Project Development**	**Knowledge:** Understanding the lifecycle of software development from requirements to deployment. **Skills:** Deliver end-to-end solutions that meet user needs. **Attitudes:** Integrate frontend and backend components to create cohesive applications.

3. Syllabus

FULLSTACK DEVELOPMENT SEMESTER – VI			
Course Code	**M23BCS601**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(2:2:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 Hours Theory +20 Hours Practical**	Total Marks	**100**
Credits	**4**	Exam Hours	**03**

Course Learning Objectives:

1. Explain the use of learning full stack web development.
2. Make use of rapid application development in the design of responsive web pages.
3. Illustrate Models, Views and Templates with their connectivity in Django for full stack web development.
4. Demonstrate the use of state management and admin interfaces automation in Django.
5. Design and implement Django apps containing dynamic pages with SQL databases.

Module-1: MVC based Web Designing

Web framework, MVC Design Pattern, Django Evolution, Views, Mapping URL to Views, Working of Django URL Confs and Loose Coupling, Errors in Django, Wild Card patterns in URLS.

Textbook 1: Chapter 1 and Chapter 3

Module -2: Django Templates and Models

Template System Basics, Using Django Template System, Basic Template Tags and Filters, MVT Development Pattern, Template Loading, Template Inheritance, MVT Development Pattern.

Configuring Databases, Defining and Implementing Models, Basic Data Access, Adding Model String Representations, Inserting/Updating data, Selecting and deleting objects, Schema Evolution

Textbook 1: Chapter 4 and Chapter 5

Module -3: Django Admin Interfaces and Model Forms

Activating Admin Interfaces, Using Admin Interfaces, Customizing Admin Interfaces, and Reasons to use Admin Interfaces.

Form Processing, Creating Feedback forms, Form submissions, custom validation, creating Model Forms, URLConf Ticks, and Including Other URLConfs.

Textbook 1: Chapters 6, 7 and 8

Module -4: Generic Views and Django State Persistence

Using Generic Views, Generic Views of Objects, Extending Generic Views of objects, Extending Generic Views.

MIME Types, Generating Non-HTML contents like CSV and PDF, Syndication Feed Framework, Sitemap framework, Cookies, Sessions, Users and Authentication.

Textbook 1: Chapters 9, 11 and 12

Module -5: jQuery and AJAX Integration in Django

Ajax Solution, Java Script, XHTML HttpRequest and Response, HTML, CSS, JSON, iFrames, Settings of Java Script in Django, jQuery and Basic AJAX, jQuery AJAX Facilities, Using jQuery UI Autocomplete in Django

Textbook 2: Chapters 1, 2 and 7.

Laboratory Component:

1. Installation of Python, Django and Visual Studio code editors can be demonstrated.Creation of virtual environment, Django project and App should be demonstrated

2. Develop a Django app that displays

A.current date and time in server

B. date and time four hours ahead and four hours before as an offset of current date and time in server.

3. Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event

4. Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages: contact us, About Us and Home page of any website.

5. Develop a Django app that performs student registration to a course.

A. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.

B. For student and course models created, register admin interfaces, perform migrations and illustrate data entry through admin forms.

C..For students enrolment, create a generic class view which displays list of students and detail view that displays student details for any selected student in the list.

6. Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project.

9.. Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.

10. Develop a registration page for student enrolment as done in Program 5, but without page refresh using AJAX.

11. Develop a search application in Django using AJAX that displays courses enrolled by a student being searched.

Textbooks :

1. Adrian Holovaty, Jacob Kaplan Moss, The Definitive Guide to Django: Web Development Done Right, Second Edition, Springer-Verlag Berlin and Heidelberg GmbH & Co. KG Publishers, 2009

2. Jonathan Hayward, Django Java Script Integration: AJAX and jQuery, First Edition, Pack Publishing, 2011

Reference Books:

1. Aidas Bendroraitis, Jake Kronika, Django 3 Web Development Cookbook, Fourth Edition, Packt Publishing, 2020

2. William Vincent, Django for Beginners: Build websites with Python and Django, First Edition, Amazon Digital Services, 2018

3. Antonio Mele, Django3 by Example, 3rd Edition, Pack Publishers, 2020

Weblinks and Video Lectures (e-Resources):

1. MVT architecture with Django: https://freevideolectures.com/course/3700/django-tutorials

2. Using Python in Django: https://www.youtube.com/watch?v=2BqoLiMT3Ao

3. Model Forms with Django: https://www.youtube.com/watch?v=gMM1rtTwKxE

4. Real time Interactions in Django: https://www.youtube.com/watch?v=3gHmfoeZ45k

5. AJAX with Django for beginners: https://www.youtube.com/watch?v=3VaKNyjlxAU

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving - applying the Django framework concepts and its integration with AJAX to develop any shopping website with admin and user dashboards.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1: MVC based Web Designing	Understand the concept of MVC and MVT architecture, Evolution of Django, views, working of Django URL confs and Loose coupling. Understand how to install python,Django and visual studio code and how to create virtual environment and develop a Django app that display current date and time with different scenarios
2	Week 2: Django Templates	Creating HTML templates, Using Django's template language for dynamic content, Template inheritance and context rendering. Develop a simple Django app of order and unorder list of fruits and students for an event and also develop an app for layout.html
3	Week 3-4: Models	Defining Django models and fields, Performing database migrations. Introduction to Django's ORM (Object-Relational Mapping),understanding concept of inserting, updating ,selecting and deleting the data in database Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.
4	Week 5-6: Django Admin Interfaces	Understand the concept of admin interface and how to use, active and customizing admin interface.

		For student and course models, register admin interfaces, perform migrations and illustrate data entry through admin forms.
5	Week 7-8: Model Forms	Creating and handling forms in Django and understand how to create a feedback form, model forms. Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project
6	Week 9-10: Generic Views	Definition of generic views, extending generic views of object, different types of generic view. Understanding the concept of generating non-HTML contents like CSV and PDF. For student's enrolment, create a generic class view which displays list of students and detail view that displays student details for any selected student in the list. Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.
7	Week 11-12: jQuery and AJAX Integration in Django	Understand the concept of HTML request and Response,jQuery,Ajax solution, CSS, JSON with code snippet Develop a registration page for student enrolment but without page refresh using AJAX. Develop a search application in Django using AJAX that displays courses enrolled by a student being searched.

5. Teaching-Learning Process Strategies

S/L	**TLP Strategies:**	**Description**
1	Interactive Lectures and Demos	Use interactive lectures to introduce new concepts, supported by live coding sessions where the instructor demonstrates how to implement these concepts in real-time. For example, while teaching Django models, the instructor can create models in a live coding session and show how they translate into database tables.
2	Hands-on Labs and Coding Exercises	Incorporate regular lab sessions where students work on coding exercises that apply the concepts taught in lectures. After learning about Django views and templates, assign exercises where students create their own views and templates for a small web application.
3	Project-Based Learning	Use a project-based learning approach where students build a full stack application over the course duration, integrating all aspects of development (front-end, back-end, database, and deployment). Start with smaller individual projects (e.g., a simple blog), leading up to a capstone project where students create a more complex application, such as an e-commerce site.
4	Collaborative Learning and Pair Programming	Encourage collaborative learning through pair programming and group projects, where students can work together to solve problems. Assign pair programming tasks, such as creating RESTful APIs or implementing authentication, and group projects where teams build different modules of a larger application.
5	Flipped Classroom Approach	Implement a flipped classroom model, where students study theoretical concepts through online resources (videos, articles) before class, and class time is dedicated to discussions, Q&A, and practical exercises. Assign pre-class videos on Django's ORM, and use class time to discuss advanced queries and perform hands-on exercises.
6	Continuous Assessment and Feedback	Use formative assessments (quizzes, coding challenges) and provide regular feedback to monitor student progress and address learning gaps. After each module, conduct a quiz or a coding challenge to assess understanding. Provide detailed feedback on code quality, efficiency, and best practices.
7	Use of Real-World Case Studies	Integrate real-world case studies into the curriculum to demonstrate how full stack development is applied in various industries. Analyze a case study of a successful web application, discussing the technologies used, the architecture, and the challenges faced during development.
8	Guest Lectures and Industry Interaction	Invite industry professionals to give guest lectures or hold Q&A sessions to provide insights into the latest trends and best practices in full stack development.

		Arrange guest lectures on topics like cloud deployment, scalability, or emerging technologies in full stack development.
9	Supplementary Resources and Self-Learning	Provide supplementary resources like online tutorials, documentation, and recommended reading to encourage self-learning and deeper exploration of topics. Share resources on advanced Django topics, front-end frameworks, or best practices in web security, encouraging students to explore these on their own.

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**
	Total Marks		**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
	Total Marks		**100%**	**25**	**10**

Final CIE Marks = (A) + (B)

Semester End Examination pattern:

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 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. 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	Master Core Python Programming Skills	Develop a solid understanding of Python programming, including advanced concepts such as object-oriented programming, file handling, and libraries.
2	Understand and Apply Django Framework	Gain in-depth knowledge of Django’s MVC (Model-View-Controller) architecture and how it facilitates web development.
3	Develop Database Management Skills	Learn how to design, implement, and manage databases using Django's ORM and SQL.
4	Build and Style User Interfaces	Acquire skills in front-end technologies like HTML, CSS, and JavaScript to create responsive and interactive user interfaces.
5	Implement Security Best Practices	Learn how to secure web applications using Django’s built-in security features and web security best practices.
6	Integrate Front-End Frameworks with Django	Learn how to integrate modern front-end frameworks (e.g., React, Vue.js) with Django for building dynamic, single-page applications
7	Test and Debug Django Applications	Understand the importance of testing and debugging, and learn how to write and run tests in Django

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

Course Outcomes (COs)

COs	Description
M23BCS601.1	Describe the working of MVT based full stack web development with Django.
M23BCS601.2	Apply the Django framework libraries to render non HTML contents like CSV and PDF.
M23BCS601.3	Analyse the role of Template Inheritance and Generic views for developing full stack web applications
M23BCS601.4	Designing of Models and Forms for rapid development of web pages and Perform jQuery based AJAX integration to Django Apps to build responsive full stack web applications.
M23BCS601.5	Demonstrate the ability to design and develop python programs by using various Django frameworks and document the result.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS601.1	3	-	-	-	-	-	-	-	-	-	-	2	2	-
M23BCS601.2	-	3	-	-	-	-	-	-	-	-	-	2	2	-
M23BCS601.3	-	3	2	-	-	-	-	-	-	-	-	2	-	2
M23BCS601.4	-	-	2	-	2	-	-	-	-	-	-	2	-	3
M23BCS601.5	3	3	3	-	3	-	-	-	-	-	-	2	3	3
M23BCS601	**3**	**3**	**2.3**	**-**	**2.5**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2.33**	**2.33**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**10**					**10**
Module 2		**4**	**6**			**10**
Module 3		**10**				**10**
Module 4			**5**	**5**		**10**
Module 5					**10**	**10**
Total	**10**	**14**	**10**	**5**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Module 1	**20**					**20**
Module 2		**8**	**12**			**20**
Module 3		**20**				**20**
Module 4			**10**	**10**		**20**
Module 5					**20**	**20**
Total	**20**	**28**	**22**	**10**	**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

1. Continued Popularity and Growth

- Widespread Adoption: Django will continue to be a popular choice for developers due to its robustness, ease of use, and the large number of built-in features that simplify complex tasks.
- Community and Ecosystem: The Django community is active and growing, with ongoing development of new features, libraries, and extensions, ensuring that Django remains relevant and up-to-date.

2. Enhanced Performance and Scalability

- Machine Learning and AI: Integration with machine learning libraries and AI tools will become more prevalent, allowing developers to incorporate intelligent features into their Django applications.
- Blockchain: Django could be used to develop applications that interact with blockchain technologies, providing secure and decentralized solutions.

3. Advanced Front-End Integration

- Single-Page Applications (SPAs): Django will increasingly integrate with modern front-end frameworks like React, Vue.js, and Angular to build dynamic SPAs, improving user experience and application responsiveness.
- Progressive Web Apps (PWAs): Django’s ability to support PWAs will allow developers to create applications that work seamlessly across different platforms and devices, offering an app-like experience in the browser.

4. Enhanced Security Features

- **Built-In Security Enhancements:** As security threats evolve, Django will continue to strengthen its security features to protect against new vulnerabilities and ensure secure web application development.
- **Compliance:** Django will support compliance with various data protection regulations and standards, making it easier for developers to build applications that meet legal requirements

5. Growing Job Market and Career Opportunities

- Increased Demand: The demand for full stack developers skilled in Django will likely continue to grow, driven by the need for robust web applications across various industries.
- Career Specialization: Developers with expertise in Django will have opportunities to specialize in areas such as enterprise application development, e-commerce solutions, and data-driven applications.

6. Cloud and DevOps Integration

- Cloud-Native Development: Django applications will increasingly be designed for cloud environments, leveraging cloud services for scalability, storage, and computing power.
- DevOps Practices: Integration with DevOps practices will streamline the development, deployment, and maintenance of Django applications, promoting continuous integration and continuous deployment (CI/CD) workflows.

6thSemester	**Professional Core Course (PC)** **Machine Learning**	**M23BCS602**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Mathematics	• Good understanding of calculus (derivatives, integrals, optimization). • Familiarity with probability and statistics (probability distributions, hypothesis testing).
2.	Programming	• Proficiency in at least one programming language commonly used for machine learning (Python, R). • Experience with data structures and algorithms. • Basic understanding of software engineering principles (code organization, debugging).
3.	Basic Understanding of Data Structures	• Concepts of arrays, lists, trees, and basic algorithms for sorting and searching.
4.	Foundation in linear algebra	• Knowledge of matrices, vectors, eigenvalues, gradients, and optimization.

2. Competencies

S/L	Competency	KSA Description
1	Data Exploration and Preparation	• **Knowledge:** Understanding various data types, data quality issues, and data preprocessing techniques. • **Skills:** Ability to clean, transform, and prepare data for machine learning algorithms. • **Attitudes:** Appreciation for the importance of high-quality data.
2	Supervised Learning Algorithms	• **Knowledge:** Understanding the principles of supervised learning, including regression and classification algorithms (linear regression, logistic regression, decision trees, support vector machines). • **Skills:** Ability to apply, evaluate, and tune these algorithms. • **Attitudes:** Critical thinking in model selection and evaluation.
3	Unsupervised Learning and Dimensionality Reduction	• **Knowledge:** Understanding clustering and dimensionality reduction techniques (k-means, PCA). • **Skills:** Ability to apply these methods for data exploration and feature engineering. • **Attitudes:** Openness to exploring data patterns and reducing data complexity.
4	Model Evaluation and Selection	• **Knowledge:** Understanding metrics for evaluating model performance (accuracy, precision, recall, F1-score, ROC curves). • **Skills:** Ability to compare models, perform cross-validation, and select the best model for a given task. • **Attitudes:** Objectivity in model assessment.
5	Ethical Considerations in Machine Learning	• **Knowledge:** Awareness of ethical implications, bias in data, and fairness considerations. • **Skills:** Ability to identify and mitigate bias in machine learning models. • **Attitudes:** Responsibility and a commitment to ethical AI practices.

3. Syllabus

Machine Learning **SEMESTER – VI**			
Course Code	**M23BCS602**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**50 Hours**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**
Course Objectives: 1. Introduce the fundamental concepts and principles of machine learning.			

2. Provide hands-on experience with commonly used machine learning algorithms.
3. Develop skills in data preprocessing, feature engineering, model building, evaluation, and selection.
4. Foster critical thinking about the ethical implications of machine learning.
5. Prepare students to apply machine learning techniques to real-world problems.

Module -1 **(10 hours)**

Machine Learning and Data Essentials

Introduction to Machine Learning, Machine Learning Explained, Machine Learning in Relation to Other Fields, Types of Machine Learning, Challenges of Machine Learning, Machine Learning Process, Machine Learning Applications, What is Data?, Big Data Analytics and Types of Analytics, Big Data Analysis Framework.

Chapter 1 (All Sections), Chapter 2 (Sections 2.1, 2.2, 2.3)

Module -2 **(10 hours)**

Data Analysis and Learning Foundations.

Descriptive Statistics, Univariate Data Analysis and Visualization, Bivariate Data and Multivariate Data, Essential Mathematics for Multivariate Data, Introduction to Learning and its Types.

Chapter 2 (Sections 2.4, 2.5, 2.6, 2.7, 2.8), Chapter 3 (Section 3.1)

Module -3 **(10 hours)**

Regression and Supervised Learning

Introduction to Computation Learning Theory, Design of a Learning System, Introduction to Concept Learning, Induction Biases, Modeling in Machine Learning, Learning Frameworks, Introduction to Regression, Introduction to Linearity, Correlation, and Causation, Introduction to Linear Regression, Validation of Regression Methods.

Chapter 3 (Sections 3.2, 3.3, 3.4, 3.5, 3.6, 3.7), Chapter 5 (Sections 5.1, 5.2, 5.3, 5.4)

Module -4 **(10 hours)**

Advanced Supervised Learning and Bayesian Principles

Multiple Linear Regression, Polynomial Regression, Logistic Regression, Ridge, Lasso, and Elastic Net Regression, Introduction to Decision Tree Learning Model, Decision Tree Induction Algorithms, Introduction to Probability-based Learning, Fundamentals of Bayes Theorem, Classification Using Bayes Model, Introduction to Ensemble Learning, Ensembling Techniques, Parallel Ensemble Models.

Chapter 5 (Sections 5.5, 5.6, 5.7, 5.8), Chapter 6 (Sections 6.1, 6.2), Chapter 8 (Sections 8.1, 8.2, 8.3), Chapter 12 (Sections 12.1, 12.2).

Module -5 **(10 hours)**

Unsupervised Learning, Reinforcement Learning, and Beyond

Introduction to Clustering Approaches, Proximity Measures, Hierarchical Clustering Algorithms, Partitional Clustering Algorithm, Overview of Reinforcement Learning, Scope of Reinforcement Learning, Reinforcement Learning As Machine Learning, Probabilistic Graphical Models, Artificial Neural Networks, Support Vector Machines, Genetic Algorithms, Deep Learning.

Chapter 13 (Sections 13.1, 13.2, 13.3, 13.4), Chapter 14 (Sections 14.1, 14.2, 14.3), Overview of Chapters **9, 10, 11, 15, 16.**

TEXTBOOK:

1. S. Sridhar, M Vijayalakshmi "Machine Learning". Oxford ,2021
2. Tom Michel, Machine Learning, McGrawHill Publication.

REFERENCE BOOKS:

1. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020.
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 1st Edition, 2006.

VIDEO LINKS:

1. Machine Learning Course by Andrew Ng (Stanford CS229): https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU
2. 3Blue1Brown - Essence of Linear Algebra & Calculus:
   Linear Algebra: https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab
   Calculus: https://www.youtube.com/playlist?list=PLZHQObOWTQDMsr9K-rj53DwVRMYO3t5Yr
3. StatQuest with Josh Starmer (YouTube Channel): https://www.youtube.com/user/joshstarmer

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	Introduction to ML, Types of Learning, Applications, Data Types, Descriptive Statistics, Visualization (Modules 1 & 2, introductory portions)
2	Week 3-4:	Regression Analysis: Linear, Multiple Linear, Polynomial, Model Evaluation (Module 3)
3	Week 5-6:	Classification with Logistic Regression, Regularization, Bias-Variance Tradeoff (Module 3 & 4 - introductory part)
4	Week 7-8:	Decision Trees, Ensemble Methods (Bagging, Random Forests), Evaluating Classifiers (Module 4)
5	Week 9-10:	Unsupervised Learning: Clustering (k-means), Dimensionality Reduction (PCA), Applications (Module 5)
6	Week11-12:	Reinforcement Learning Fundamentals, Applications, Introduction to Advanced Topics: Probabilistic Graphical Models, Neural Networks, Support Vector Machines, Genetic Algorithms, Deep Learning (Module 5 & Overviews)

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Lecture Method	Deliver core concepts and foundational knowledge, interactive discussions.
2	Code Demonstrations	In-class coding examples using Python and relevant ML libraries (like scikit-learn) to illustrate algorithm implementation and data analysis techniques.
3	Project-Based Learning	Hands-on projects where students apply ML to datasets and solve real-world inspired problems.
4	Case Studies	Analyze real-world applications of machine learning to understand its impact and challenges.
5	Group Discussions	Facilitate critical thinking and encourage collaborative learning through discussions on ethical considerations, future trends, and current events in AI/ML.
6	Guest Lectures	Invite industry experts to provide insights into practical applications and career paths in Machine Learning.

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

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 mark

7. Learning Objectives

S/L	Learning Objectives	Description
1	Data Understanding and Preparation for Machine Learning	Comprehend data types, data quality issues, and learn data preprocessing techniques for effective machine learning.
2	Supervised Learning Concepts and Algorithms	Understand the principles of supervised learning, including regression and classification algorithms. Be able to apply, evaluate, and tune these algorithms for predictive modeling.
3	Unsupervised Learning Techniques	Grasp the concepts of clustering, dimensionality reduction, and their applications in data exploration and feature engineering. Be able to apply common methods like k-means and PCA.
4	Model Evaluation, Selection, and Performance Metrics	Learn various metrics for evaluating model performance and be able to compare and select the best model for a given task using techniques like cross-validation.
5	Ethical Implications and Applications of Machine Learning	Develop an understanding of the ethical implications of machine learning, including bias in data, fairness considerations, and responsible AI development practices.
6	Future Trends and Advanced Topics in Machine Learning	Gain awareness of current trends, emerging technologies, and advanced topics in machine learning to foster continuous learning and exploration in the field.

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

Course Outcomes (COs)

COs	Description
M23BCS602.1	Understand the fundamental concepts of machine learning, its types, applications, and the ethical implications of its use.
M23BCS602.2	Apply data preprocessing techniques and perform exploratory data analysis to prepare data for machine learning algorithms.
M23BCS602.3	Implement unsupervised learning techniques for clustering and dimensionality reduction.
M23BCS602.4	Build, evaluate, and compare supervised learning models for regression and classification tasks.
M23BCS602.5	Analyze and interpret the results of machine learning models and communicate findings effectively.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS602.1	2	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCS602.2	-	3	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS602.3	-	-	3	-	3	-	-	-	-	-	-	-	-	3
M23BCS602.4	-	-	-	3	3	-	-	-	-	-	-	-	3	-
M23BCS602.5	**-**	**-**	-	3	-	-	-	-	-	3	-	-	-	3
M23BCS602	**2**	**3**	**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**

10. Future with this Subject:

1. **Deep Learning Specialization:** The knowledge gained here primes you to explore the fascinating world of deep learning (CNNs, RNNs, Transformers). Imagine designing algorithms for self-driving cars, medical image analysis, or even writing creative text—all made possible with deep learning and highly relevant to your future careers.
2. **MLOps (Machine Learning Operations):** Building ML models is just the start. MLOps focuses on deploying, managing, and scaling those models to real-world applications, a crucial skill for ML engineers and data scientists.
3. **Explainable AI (XAI):** As AI makes more decisions, understanding why becomes paramount. XAI techniques, which focus on interpreting model behavior, are essential for building trust and ensuring ethical AI development, especially in sensitive fields like healthcare and finance.
4. **Edge Computing and IoT:** The future is interconnected. By integrating ML with edge devices and the Internet of Things (IoT), you can create intelligent systems that react in real-time – from smart homes and factories to autonomous vehicles, creating massive career opportunities in the process.
5. **Reinforcement Learning Applications:** Imagine teaching robots to learn like humans. Reinforcement learning makes this possible and is poised to revolutionize fields like robotics, control systems, and personalized learning experiences.
6. **Generative AI:** This rapidly advancing field uses models like GANs and diffusion models to create stunningly realistic images, videos, and even music, blurring the lines between human and machine creativity and opening exciting career paths for those with the right ML skills.

6th Semester	Professional Elective-II (PE) BLOCKCHAIN TECHNOLOGY	M23BCS603A

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Programming Skills	Understanding of programming concepts and knowledge of languages like Python, JavaScript, or C++ is essential.
2	Data Structures	Familiarity with data structures such as linked lists, hash maps, and graphs is crucial.
3	Basic Knowledge of Computer Security	A good grasp of cryptographic principles is important for securing blockchain transactions.
4	Distributed Systems and Networking	Understanding how distributed systems work and the basics of networking can help grasp how blockchain nodes communicate and maintain consensus.
5	Basic Knowledge of Economics	Since blockchain often intersects with cryptocurrencies, having a basic understanding of economic principles can be beneficial.

2. Competencies

S/L	Competency	KSA Description
1	**Blockchain Fundamentals**	**Knowledge:** Understand the basic concepts of blockchain, including its architecture, how it works, and its various applications. **Skills:** Explain blockchain concepts clearly and apply them in practical scenarios. **Attitudes:** Curiosity and eagerness to explore new technologies and their potential impacts.
2	**Decentralized system & Cryptography**	**Knowledge:** Familiarity with decentralized networking and cryptographic principles such as public-key cryptography, cryptographic hashing, and digital signatures. **Skills:** Implement cryptographic algorithms like Secret Key Cryptography (SKC), Public Key Cryptography (PKC) & Hash Functions. **Attitudes:** Attention to detail and a strong focus on security and privacy.
3	**Basic Knowledge of Cryptocurrency**	**Knowledge:** • Understand the basic principles of Bitcoin, including its creation, blockchain technology, mining process, and transaction mechanisms. • Familiarity with various alternative cryptocurrencies like Ethereum, Litecoin, Ripple, and others, including their unique features and use cases. **Skills:** • Proficiency in programming languages such as Python, Solidity, and JavaScript for developing blockchain applications and smart contracts. • Understanding and implementing security measures to protect digital wallets, private keys, and transactions from potential threats. **Attitudes:** • Commitment to ethical practices in cryptocurrency development.
4	**Smart Contracts**	**Knowledge:** Understanding of smart contracts, particularly on platforms like Ethereum, and how they automate and enforce agreements. **Skills:** Proficiency in writing and deploying smart contracts using languages like python. **Attitudes:** Precision and thoroughness in coding to ensure accuracy and reliability.
5	**Understanding Different Blockchains**	**Knowledge:** Understanding of non-Cryptocurrency Applications like supply chain management, IoT and others. **Skills:** Proficiency in blockchain fundamentals. **Attitudes:** Willingness to adapt to the rapidly changing landscape of blockchain.

3. Syllabus

BLOCKCHAIN TECHNOLOGY
SEMESTER – VI

Course Code	**M23BCS603A**	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.Define and explain the fundamentals of Blockchain
2.Illustrate the technologies of blockchain
3.Describe the models of blockchain
4.Analyze and demonstrate the Ethereum

Module -1

Blockchain 101: Distributed systems, History of blockchain, Introduction to blockchain, Types of blockchain, CAP theorem and blockchain, Benefits and limitations of blockchain.
Text Book 1: Chapter 1

Module -2

Decentralization and Cryptography: Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Decentralized organizations. Cryptography and Technical Foundations: Cryptographic primitives, Asymmetric cryptography, Public and private keys
Text Book 1: Chapter 2, Chapter 4

Module -3

Bitcoin
Bitcoin, Transactions, Blockchain, Bitcoin payments. Theoretical foundations, Bitcoin limitations, Namecoin, Litecoin, Primecoin, Zcash
Text Book 1: Chapter 5, Chapter 6, Chapter 8

Module -4

Smart Contracts and Ethereum 101:
Smart Contracts: Definition, Ricardian contracts. Ethereum 101: Introduction, Ethereum blockchain, Elements of the Ethereum blockchain, Precompiled contracts.
Text Book 1: Chapter 9, Chapter 10

Module -5

Alternative Blockchains: Blockchains
Blockchain- Outside of Currencies: Internet of Things, Government, Health, Finance, Media
Text Book 1: Chapter 17

Text Books:

1. Mastering Blockchain - Distributed ledgers, decentralization and smart contracts explained, Imran Bashir, Packt Publishing Ltd, Second Edition, ISBN 978-1-78712-544-5, 2017
2. Bitcoin and Cryptocurrency Technologies, Arvind Narayanan, Joseph Bonneau, Edward Felten, 2016

Reference Books:

1.Blockchain Basics: A Non-Technical Introduction in 25 Steps, Daniel Drescher, Apress, First Edition, 2017
2.Mastering Bitcoin: Unlocking Digital Cryptocurrencies, Andreas M. Antonopoulos, O'Reilly Media, First Edition, 2014.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2: Blockchain 101	● **Competency**: Blockchain Fundamentals ● **Knowledge**: Understand the basic concepts of blockchain, including its architecture, how it works, and its various applications. ● **Skills**: Explain blockchain concepts clearly and apply them in practical scenarios.
2	Week 3-4: Decentralization and Cryptography	● **Competency**: Decentralized system & Cryptography

3		● **Knowledge**: Familiarity with decentralized networking and cryptographic principles such as public-key cryptography, cryptographic hashing, and digital signatures. ● **Skills:** Implement cryptographic algorithms like Secret Key Cryptography (SKC), Public Key Cryptography (PKC) & Hash Functions.
3	Week 5-6: Bitcoin and Alternative Coins	● **Competency:** Basic Knowledge of cryptocurrency ● **Knowledge**: Understand the basic principles of Bitcoin, including its creation, blockchain technology, mining process, and transaction mechanisms. Familiarity with various alternative cryptocurrencies like Ethereum, Litecoin, Ripple, and others, including their unique features and use cases. ● **Skills**: Proficiency in programming languages such as Python, Solidity, and JavaScript for developing blockchain applications and smart contracts. Understanding and implementing security measures to protect digital wallets, private keys, and transactions from potential threats.
4	Week 7-8: Smart Contracts and Ethereum 101	● **Competency:** Smart Contracts ● **Knowledge:** Understanding of smart contracts, particularly on platforms like Ethereum, and how they automate and enforce agreements. ● **Skills:** Writing and deploying smart contracts using languages like python.
5	Week 9-10: Alternative Blockchains	● **Competency:** Understanding Different Blockchains ● **Knowledge**: Understanding of non-Cryptocurrency Applications like supply chain management, IoT and others. ● **Skills**: Proficiency in blockchain fundamentals.

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)

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
	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 Blockchain Technology Fundamentals	Students will grasp the fundamental concepts of blockchain and it's decentralized way of function, including smart contracts, cryptocurrencies, non-crypto blockchains
2	Developing blockchain applications	Students will learn to develop and implement blockchain applications, smart contracts and for real world scenarios using programming languages like python.
3	Project-Based Learning	Through hands-on projects, students will apply their knowledge of blockchain to design, develop and implement blockchain applications, 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 blockchain technology, including transparency and compliance with regulations, cautious and strategic approach to managing risks associated with cryptocurrency investments and trading, best practices for securing blockchain networks and applications.

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

Course Outcomes (COs)

COs	Description
M23BCS603A.1	Explain fundamentals of blockchain and how it works.
M23BCS603A.2	Implement security measures through cryptographic principles.
M23BCS603A.3	Describe fundamentals of cryptocurrencies and their role in economics.
M23BCS603A.4	Analyse and demonstrate the Ethereum.
M23BCS603A.5	Analyse and demonstrate Hyperledger fabric.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS603A.1	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS603A.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS603A.3	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS603A.4	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS603A.5	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**3**
M23BCS603A	**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 "Blockchain Technology" course in the seventh semester of the B.E program lays a strong foundation for several future opportunities in the industry. The contributions of this subject extend across various areas, enhancing the students' understanding and skills in the field of Blockchain and Cryptocurrencies. Here are some notable contributions:

- **Enhanced Security and Transparency:** Blockchain's decentralized nature ensures that data is secure and transparent, reducing the risk of fraud and tampering. In supply chain management, companies are using blockchain to track the journey of products from origin to consumer, ensuring authenticity and reducing counterfeiting.
- **Decentralized Finance (DeFi):** DeFi platforms are revolutionizing traditional financial systems by providing decentralized financial services without intermediaries.
- **Integration with Emerging Technologies:** Blockchain is being integrated with other emerging technologies like Artificial Intelligence (AI) and the Internet of Things (IoT) to create innovative solutions.
- **Enterprise Adoption:** More enterprises are adopting blockchain technology to improve efficiency, reduce costs, and enhance security. Companies like Walmart and Maersk are using blockchain for supply chain management, while financial institutions are exploring blockchain for secure and transparent transactions.
- **Regulatory Developments:** As blockchain technology matures, regulatory frameworks are evolving to provide clearer guidelines and foster innovation. Governments are exploring Central Bank Digital Currencies (CBDCs) to leverage blockchain for secure and efficient digital currencies.
- **Non-Fungible Tokens (NFTs):** NFTs are expanding beyond digital art to include various applications such as virtual real estate, gaming, and intellectual property. Platforms like Decentraland and Axie Infinity use NFTs to create virtual worlds and gaming ecosystems where users can own and trade digital assets.

6th Semester	**Professional Elective-II** **FILE STRUCTURE**	**M23BIS603B**

1. Prerequisites

S/L	**Proficiency**	**Prerequisites**
1	Understanding of Data Types and Requirements	Knowing the types of data to be stored and their requirements (e.g., text, binary data, images) helps in designing an effective file structure.
2	Organization Needs	Identifying how data will be accessed, organized, and managed. This includes understanding hierarchical, flat, or relational structures based on the use case.
3	Storage Capacity	Considering the amount of data to be stored and how it will grow over time helps in designing a scalable file structure.
4	Performance Requirements	Determining how quickly data needs to be accessed or modified impacts the file structure design. This includes indexing and retrieval efficiency.
5	File Naming Conventions	Establishing consistent naming conventions for files and directories to ensure clarity and ease of access.
6	Security and Permissions	Implementing appropriate security measures and access controls to protect data from unauthorized access or modification.
7	Backup and Recovery Plans	Designing strategies for backing up data and recovering it in case of loss or corruption.
8	Compatibility and Integration	Ensuring that the file structure is compatible with existing systems and software and can integrate seamlessly with other systems if needed.
9	Standards and Best Practices	Following industry standards and best practices for file structure design to ensure consistency and reliability.

2. Competencies

S/L	**Competency**	**KSA Description**
1	Design and Organization	**Understanding of Hierarchical and Flat Structures:** Ability to design both hierarchical (tree-like) and flat file systems based on data needs. **Data Classification:** Skills in classifying and organizing data into appropriate categories and folders.
2	File Management	**File Naming Conventions:** Knowledge of effective naming conventions to ensure consistency and clarity. **Directory Structure Management:** Competency in creating and maintaining a logical and efficient directory structure.
3	[Performance Optimization	**Indexing:** Ability to implement and manage indexing to improve data retrieval times. **Data Compression:** Knowledge of techniques to compress files and optimize storage usage.
4	Security and Permission	**Access Control:** Skills in setting up and managing file permissions and access controls to protect data. **Encryption:** Understanding of file encryption methods to secure sensitive information.
5	Backup and Recovery	**Backup Strategies:** Competency in developing and implementing effective backup strategies. **Recovery Procedures:** Knowledge of procedures for recovering data in case of loss or corruption.

6	Scalability and Growth Management	**Capacity Planning:** Ability to plan for future growth and scalability of the file system. **File System Expansion:** Skills in managing and expanding file systems as data volume increases.
7	Compatibility and Integration	**System Integration:** Knowledge of how to integrate file structures with existing systems and software. **Data Interchange:** Understanding of file formats and standards for data interchange between systems.
8	Troubleshooting and Maintenance	**Problem Solving:** Ability to diagnose and resolve issues related to file structure and management. **Routine Maintenance:** Competency in performing regular maintenance tasks to ensure the file structure remains efficient and reliable.
9	Documentation	**File Structure Documentation:** Skills in documenting file structures, conventions, and procedures for clarity and future reference.

3. Syllabus

FILE STRUCTURE
SEMESTER – VI

Course Code	**M23BIS603B**	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 T**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course Learning Objectives: This course will enable students to:
1. Explain the fundamentals of file structures and their management.
2 Measure the performance of different file structures.
3 Organize different file structures in the memory.
4. Demonstrate hashing and indexing techniques.

Module-1: Introduction

File Structures: The Heart of the file structure Design, A Short History of File Structure Design, A Conceptual Toolkit;
Fundamental File Operations: Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking**,**
Secondary Storage and System Software: Disks, Magnetic Tape, Disk versus Tape; CD-ROM: Introduction, Physical Organization, Strengths and Weaknesses;
Fundamental File Structure Concepts, Managing Files of Records: Field and Record Organization
Textbook 1: Chapters 1, 2.1-2.5, 3,.1-3.6, and 4.1

Module -2: Organization of Files for Performance, Indexing

Organization of Files for Performance, Indexing: Data Compression, Reclaiming Space in files, Internal Sorting and Binary Searching, Keysorting; What is an Index? A Simple Index for Entry-Sequenced File, Using Template Classes in C++ for Object I/O, Entry-Sequenced Files of Data Objects, Indexes that are too large to hold in Memory, Indexing to provide access by Multiple keys, Retrieval Using Combinations of Secondary Keys, Improving the Secondary Index structure: Inverted Lists
Textbook 1: Chapters 6 and 7.1-7.8

Module -3: Consequential Processing and the Sorting of Large Files

Consequential Processing and the Sorting of Large Files: A Model for Implementing Cosequential Processes, Extension of the Model to include Mutiway Merging, A Second Look at Sorting in Memory
Multi-Level Indexing and B-Trees: The invention of B-Tree, Statement of the problem, Indexing with Binary Search Trees; Multi-Level Indexing, B-Trees, Example of Creating a B-Tree
Textbook 1: Chapters 8.1-8.4and 9.1-9.6

Module -4: Indexed Sequential File Access and Prefix B + Trees

Indexed Sequential File Access and Prefix B + Trees: Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index: Separators Instead of Keys, The Simple Prefix B+ Tree and its maintenance, Index Set Block Size, Internal Structure of Index Set Blocks: A Variable-order B- Tree, Loading a Simple Prefix B+ Trees, B-Trees, B+ Trees and Simple Prefix B+ Trees in Perspective.
Textbook 1: Chapter 10

Module -5: Hashing

Hashing: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distribution, How much Extra Memory should be used?, Collision resolution by progressive overflow, Buckets, Making deletions, Other collision resolution techniques,.

Textbook1: Chapters 11 excluding 11.9

Textbooks

1. Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998. (Chapters 1 to 11)
2. K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.

Reference Books

1. K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.
2. Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.
3. Raghu Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.

Weblinks and Video Lectures (e-Resources):

1. File handling through C++ classes: https://www.tutorialspoint.com/file-handling-through-cplusplus-classes
2. YouTube - File Structures and File Handling in C++

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

1. **Real-World Case Studies helps student to**

Gain insights into practical applications and challenges.

Analyze and discuss case studies of real-world file systems or data processing systems.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1: Introduction to File Structures	Understand the concept of file structures and their importance, Types of file systems (hierarchical, flat, relational),
2	Week 2: File Organization Concepts	File organization methods (sequential, indexed, direct), Access methods and file systems
3	Week 3-4: Basic File Handling, Sorting Algorithms Overview	File operations (creation, reading, writing, deletion), File metadata and attributes, Introduction to sorting algorithms (Bubble Sort, Insertion Sort)
4	Week 5-6: Advanced Sorting Algorithms, File Structures and Indexing	Advanced sorting algorithms (QuickSort, MergeSort, HeapSort), Comparative analysis of algorithms
5	Week 7-8: File Compression and Encryption	Techniques for file compression (e.g., Huffman coding), Basic encryption methods for file security
6	Week 9-10: Handling Large Files	Techniques for managing and processing large files, File fragmentation and defragmentation
7	Week 11-12: Performance Optimization, Integration with Databases	Optimizing file structures and operations for performance, parallel processing and caching techniques

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Lectures and Demos	Use interactive lectures to introduce the concepts, supported by live coding sessions where the instructor demonstrates how to implement these concepts in real-time using C++ or other programming languages. For example, while teaching, the instructor creates the database models by giving various structures.
2	Hands-on Labs and Coding Exercises	Conduct hands-on labs where students can implement and experiment with file handling, sorting algorithms, and indexing techniques in programming languages such as Python, Java, or C++.

3	Project-Based Learning	Use a project-based learning approach where students design and implement file systems or manage large datasets or to create their own file system designs or optimize file operations based on specific requirements.
4	Interactive Tools and Simulations	Encourage usage of tools or simulations to demonstrate file structures and sorting algorithms in action. Incorporate online tools that allow students to visualize file organization, indexing, and sorting processes.
5	Flipped Classroom Approach	Implement a flipped classroom model, where students study theoretical concepts through online resources (videos, articles) before class, and class time is dedicated to discussions, Q&A, and practical exercises.
6	Continuous Assessment and Feedback	Use formative assessments (quizzes, coding challenges) and provide regular feedback to monitor student progress and address learning gaps. After each module, conduct a quiz or a coding challenge to assess understanding. Provide detailed feedback on code quality, efficiency, and best practices.
7	Use of Real-World Case Studies	Integrate real-world case studies into the curriculum to demonstrate how file structure is applied to solve file-related problems. Analyze a case study of a successful web application, discussing the technologies used, the architecture, and the challenges faced during development.
8	Guest Lectures and Industry Interaction	Invite industry professionals to give guest lectures or hold Q&A sessions to provide insights into the latest trends and best practices in file structure. Arrange guest lectures on topics like consequential processing, indexing, and hashing techniques.
9	Supplementary Resources and Self-Learning	Provide supplementary resources like online tutorials, documentation, and recommended reading to encourage self-learning and deeper exploration of topics.

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
	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	Understand Basic Concepts and Terminology	Students will be able to define and explain fundamental concepts related to file structures, including file organization, access methods, and file systems.
2	Analyze and Compare File Organization Methods	Students will be able to analyze various file organization methods and compare their advantages and disadvantages.
3	Implement File Handling Operations	Learn to implement file handling operations, including creation, reading, writing, and deletion of files using programming languages.
4	Apply Sorting Algorithms to File Data	Acquire skills to implement and apply various sorting algorithms to file data and analyze their efficiency.
5	Design and Implement Indexing Techniques	Learn how to to design and implement indexing techniques to improve file access and retrieval.
6	Handle Large Files and Data Sets	Learn how to manage and process large files and datasets effectively.
7	Optimize File Systems for Performance	Analyze and optimize file systems for better performance and efficiency.
8	Develop Problem-Solving and Analytical Skills	Develop problem-solving and analytical skills through practical exercises and projects related to file structures.

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

Course Outcomes (COs)

COs	Description
M23BIS604B.1	Identify the appropriate concept of file structure design and secondary storage devices.
M23BIS604B.2	Apply appropriate designs for storage and data manipulation with object oriented programming.
M23BIS604B.3	Examine cosequential processing and sorting of files using indexed methods.
M23BIS604B.4	Analyze the hashing technique for data handling.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BIS604B.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BIS604B.2	3	-	-	-	-	-	-	-	-	-	-	-	-	3
M23BIS604B.3	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BIS604B.4	-	3	-	-	-	-	-	-	-	-	-	-	-	2
M23BIS604B	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**	**20**	**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**	**40**	**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 field of file structures is integral to the broader domains of data management, software development, and system design. As technology evolves, the role of file structures continues to expand and adapt. Here's a look at the future directions and trends related to file structures:

1. **Enhanced Data Management and Storage:**

Trend: As data volumes grow exponentially, advanced file structures will be crucial for efficient data management and storage.

Future Direction: Development of more sophisticated file systems that support large-scale data storage solutions, including distributed and cloud-based file systems.

2. Integration with Big Data Technologies:

Trend: Big data technologies such as Hadoop and Spark require efficient file structures for processing and managing vast amounts of data.

Future Direction: Evolution of file structures to optimize for big data frameworks, including file formats like Parquet and ORC that are designed for high-performance data processing.

3. File Structures in Cloud Computing:

Trend: Cloud computing platforms are increasingly used for data storage and processing, necessitating new file structures that handle cloud-specific challenges.

Future Direction: Development of file structures optimized for cloud environments, including distributed file systems and object storage solutions.

4. File Structures for Real-Time Processing:

- **Trend:** Real-time data processing applications require file structures that can handle rapid data ingestion and retrieval.
- **Future Direction:** Implementation of file structures that support low-latency access and high-throughput processing for real-time applications.

5. Enhanced Security and Privacy:

Trend: With increasing concerns about data security and privacy, file structures need to incorporate advanced security measures.

Future Direction: Integration of encryption, access control, and auditing mechanisms directly into file structures to enhance data security.

6thSemester	**Professional Elective Advanced Web Programming**	**M23BCS603C**

1. **Prerequisites**

S/L	**Proficiency**	**Prerequisites**
1	Basic Web Development Skills	Ability in HTML includes the ability to use tags, elements, and attributes effectively, as well as a solid understanding of semantic HTML for proper document structure. In CSS, one should be knowledgeable about syntax, selectors, and properties, and familiar with layout techniques such as the box model, positioning, and basic responsive design. Additionally, a basic understanding of JavaScript syntax is essential, including variables, data types, operators, and control structures like loops and conditionals, along with experience in DOM manipulation and event handling.
2	Intermediate Web Development Skills	Skill in CSS includes expertise in Flexbox and Grid Layout for creating complex layouts, as well as the ability to design responsive web pages using media queries. Experience with CSS preprocessors like SASS or LESS is also important. In JavaScript, familiarity with ES6+ features such as arrow functions, template literals, destructuring, and modules is essential, along with an understanding of asynchronous programming concepts like callbacks, promises, and async/await. Additionally, having basic experience with JavaScript libraries or frameworks enhances overall development skills.
3	Front-End Frameworks and Libraries	Introductory knowledge of at least one front-end framework or library (e.g., React, Vue.js, Angular). Understanding of fundamental concepts such as components, state management, and routing.
4	Back-End Fundamentals	Basic knowledge of server-side programming concepts and at least one server-side language (e.g., Node.js, Python, PHP) Understanding of basic database concepts and operations (CRUD) for SQL databases. Familiarity with NoSQL databases and their use cases.
5	Development Tools	IDE/Editor: Experience with an Integrated Development Environment (IDE) or text editor (e.g., VSCode, WebStorm). Command Line: Comfort with basic command-line operations.
6	APIs and Microservices (Introductory Knowledge)	Basic API Interaction: Understanding of RESTful principles and how to interact with APIs using HTTP methods (GET, POST, PUT, DELETE). Basic Authentication: Familiarity with basic authentication mechanisms (sessions, tokens).

2. **Competencies**

S/L	**Competency**	**KSA Description**
1	**Java Script**	**Knowledge:** Core Syntax & Concepts: Variables, functions, objects, async programming. Web APIs & DOM: DOM manipulation, Fetch API, RESTful services. Libraries & Frameworks: React, Angular, Node.js. Development Tools: npm, Git, Webpack. **Skills:** Problem Solving & Debugging: Analyzing code, troubleshooting. Responsive Design: Building dynamic, responsive websites. Asynchronous Programming: Handling async tasks with promises/async-await. Testing: Unit testing with Jest, Mocha. **Abilities:** Adaptability: Learning new frameworks/libraries quickly. Attention to Detail: Writing clean, efficient code. Logical Thinking: Structuring algorithms and data flows. Communication: Explaining technical concepts clearly.
2	**CSS**	**Knowledge:** Core Concepts: Selectors, specificity, box model, units (px, em, rem).

		Layout Techniques: Flexbox, Grid, positioning, floats. Styling: Colors, typography, transitions, animations. Preprocessors: SASS, LESS. Responsive Design: Media queries, mobile-first design. **Skills:** Layout Design: Creating complex layouts with Flexbox and Grid. Responsive Design: Building adaptive, cross-device layouts. Cross-Browser Compatibility: Ensuring consistent appearance across browsers. Performance Optimization: Efficient CSS, minimizing reflows/repaints. **Abilities:** Attention to Detail: Crafting precise and consistent UI elements. Creativity: Designing visually appealing and user-friendly interfaces. Problem-Solving: Debugging layout and styling issues. Adaptability: Keeping up with new CSS features and best practices.
3	**Introduction to Angular JS**	**Knowledge:** Core Concepts: Data binding, directives, modules, services, dependency injection. MVC Architecture: Understanding Angular's MVC structure. Components: Creating reusable components and custom directives. Routing: Managing navigation with Angular's routing module. HTTP Services: Interacting with RESTful APIs using $http or HttpClient. **Skills:** Application Structure: Building scalable, modular applications. Data Binding: Implementing one-way/two-way data binding. State Management: Managing application state effectively. Testing: Writing unit tests for components and services using Jasmine and Karma. **Abilities:** Adaptability: Transitioning to newer Angular versions and frameworks. Problem-Solving: Debugging complex AngularJS applications. Collaboration: Working in teams using version control and best practices. Efficiency: Optimizing performance in large-scale AngularJS apps.
4	**Introduction to Node JS Setup**	**Knowledge:** Core Concepts: Event-driven architecture, non-blocking I/O, modules. Server-Side Development: HTTP, Express.js, RESTful APIs. Package Management: npm, package.json, dependency management. Asynchronous Programming: Callbacks, Promises, async/await. Database Integration: Connecting with databases (MongoDB, MySQL) **Skills:** Server Development: Building and maintaining RESTful APIs with Express.js. Real-Time Applications: Implementing WebSockets, real-time data handling. Asynchronous Code: Managing async operations effectively. Security: Implementing authentication, authorization, and data protection. **Abilities:** Scalability: Designing scalable server-side applications. Problem-Solving: Debugging and optimizing server performance. Adaptability: Learning and integrating new Node.js libraries and tools. Collaboration: Working effectively in full-stack development teams.
5	**Database Programming with Node JS and MongoDB**	**Knowledge:** Core Concepts: Understanding of NoSQL databases, document-oriented data models. MongoDB Basics: CRUD operations, schema design, collections, indexing. Mongoose: Schema definition, model creation, validation, and middleware. Node.js Integration: Connecting Node.js applications to MongoDB. Aggregation Framework: Using aggregation pipelines for complex queries. **Skills:** Data Modeling: Designing and implementing efficient data schemas. Query Optimization: Writing efficient queries and using indexes. Data Operations: Performing CRUD operations with Mongoose.

		Middleware: Implementing middleware for validation and data transformation. **Abilities:** Scalability: Designing databases that scale with application growth. Problem-Solving: Debugging and resolving database-related issues. Adaptability: Learning and implementing new MongoDB features and best practices. Security: Ensuring data security with authentication, encryption, and access control.

3. Syllabus

Advanced Web Programming **SEMESTER – VI**			
Course Code	M23BCS603C	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 Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**
Module -1			
Refreshing Java Script and CSS CSS syntax, benefits, Responsive design, Bootstrap introduction, Java script syntax, Java script inbuilt objects, Error handling and event handling, DOM, Asynchronous Programming			
Module -2			
Introduction to Angular JS Basics and Syntax of Angular JS, Features, Advantages, Application Structure, Basics of routes and navigation, MVC with Angular JS, Services**,** Modules, Directives, Routes, Angular JS Forms and Validations, Data binding, Creating single page website using Angular JS			
Module -3			
Introduction to Node JS Setup Node JS Environment, Package Manager, Features, Console Object, Concept of Callbacks			
Module -4			
Node JS in details Events and Event Loop, timers, Error Handling, Buffers, Streams, Work with File System, Networking with Node (TCP, UDP and HTTP clients and servers), Web Module, Debugging, Node JS REST API, Sessions and Cookies, Design patterns, caching, scalability			
Module -5			
Database Programming with Node JS and MongoDB Basics of MongoDB, Data types, Connect Node JS with MongoDB, Operations on data (Insert, Find, Query, Sort, Delete, Update) using Node JS			
Text Book and			
• Angularjs in Action ISBN 9789351198383 Ruebbelke, Wiley Publication • Node.js in Action ISBN 9789386052049 Alex Young, Bradley Meck, Mike Cantelon, Tim Oxley, Marc Harter, T.J. Holowaychuk, Nathan Rajlich, Wiley Publication **Reference Books** • Node.Js in Practice ISBN 9789351197744 Alex Young, Marc Harter, Ben Noordhuis Wiley Publication. • Pro AngularJS Freeman Apress publication • Professional Node.js By Pedro Teixeira 9781118240564 Wiley Packt Publishing			

4. Syllabus structure

S/L	**Syllabus structure**	**KS Description**
1.	Module 1: Refreshing Java Script and CSS	**Knowledge:** Understanding the fundamental concepts of JavaScript, including variables, data types, and control structures. Awareness of function declarations, expressions, and the different types of functions in JavaScript. Grasping the concept of scope and hoisting in JavaScript. **Skills:** Ability to write basic JavaScript code for common programming tasks.

Dean Academics
MIT Mysore

APPROVED
MIT MYSORE
Dean Academics

Principal
MIT Mysore

		Applying control structures and functions to solve problems. Debugging simple JavaScript programs.
2	Module 2: Introduction to Angular JS	**Knowledge:** Understanding AngularJS fundamentals, including MVC architecture, directives, services, routing, and form validation. Familiarity with creating single-page applications (SPAs) and managing dependencies through dependency injection. **Skills:** Developing dynamic, modular web applications using AngularJS with two-way data binding, custom directives, and services. Implementing routing, form validation, and user interaction in AngularJS-based SPAs.
3	Module 3: Introduction to Node JS Setup	**Knowledge:**Understanding the basics of Node.js, its architecture, and how it handles asynchronous programming. Familiarity with setting up a Node.js environment and using Node Package Manager (NPM) for managing dependencies. **Skills:**Setting up and configuring a Node.js development environment. Building and running basic server-side applications using Node.js and managing dependencies with NPM.
4	Module 4: Node JS in details	**Knowledge:** In-depth understanding of Node.js core modules, event-driven architecture, and asynchronous programming. Familiarity with building RESTful APIs, working with Express.js, and handling data storage with databases like MongoDB. **Skills:** Developing complex server-side applications using Node.js, Express.js, and integrating databases. Implementing efficient, scalable APIs and managing asynchronous operations in a Node.js environment.
5	Module 5: Database Programming with Node JS and MongoDB	**Knowledge:**Comprehensive understanding of Node.js core modules, middleware, and RESTful API development. Familiarity with Express.js, database integration, and error handling in a Node.js environment. **Skills:**Building robust server-side applications with Node.js and Express.js. Implementing database interactions, middleware, and handling asynchronous processes efficiently.

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 web 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. Syllabus Timeline

S/L	**Syllabus Timeline**	**Description**
1	Week 1-2: Refreshing Java Script and CSS	**Competency**: Proficient in fundamental JavaScript concepts and basic debugging. Advanced understanding of JavaScript and foundational CSS skills for effective web design. **Knowledge**: Grasp of variables, functions, control structures, and JavaScript scope. Knowledge of advanced JavaScript features, CSS selectors, box model, and layout techniques. **Skills**: Writing and debugging basic JavaScript code and implementing core functions and control structures. Manipulating JavaScript objects and arrays, and applying CSS to create responsive layouts with Flexbox and Grid.
2	Week 3-4: Introduction to AngularJS	**Competency**:Basic proficiency in AngularJS, including understanding its core concepts and using directives effectively. Ability to manage data binding, implement services, and set up routing in AngularJS applications. **Knowledge:** Familiarity with AngularJS architecture, modules, controllers, and built-in directives. Understanding two-way data binding, AngularJS services, and routing mechanisms for SPAs. **Skills:** Creating AngularJS modules and controllers, and applying directives for data binding and DOM manipulation. Implementing data binding, creating and using services, and configuring routing for navigation within AngularJS applications.
3	Week 5-6: Introduction to Node.js Setup	**Competency:** Basic setup and understanding of Node.js environment and its core concepts. Proficiency in utilizing core Node.js modules and setting up a basic server. **Knowledge:** Understanding Node.js architecture, event-driven programming, and NPM basics.Familiarity with Node.js core modules like http, fs, and creating a simple server. **Skills:**Setting up Node.js and using NPM for package management. Building and running a basic Node.js server and managing files with core modules.
4	Week 7-8: Node.js in Detail	**Competency:**Advanced understanding of Node.js features, including asynchronous programming and middleware. Proficiency in building and integrating RESTful APIs using Node.js. **Knowledge:** Familiarity with asynchronous operations, middleware, and error handling in Node.js. Understanding API development, integration with databases, and using frameworks like Express.js. **Skills:** Implementing advanced features such as middleware and handling asynchronous processes effectively. Developing RESTful APIs, integrating with databases, and utilizing frameworks for efficient application development.
5	Week 9-10: Database Programming with Node JS and MongoDB	**Competency:** Proficiency in using MongoDB with Node.js for data storage and retrieval. Advanced skills in optimizing database operations and managing complex data interactions. **Knowledge:**

		Familiarity with advanced MongoDB features, indexing, and performance optimization techniques. Understanding MongoDB's data model, CRUD operations, and integration with Node.js applications. **Skills:** Setting up MongoDB, performing CRUD operations, and integrating MongoDB with Node.js using Mongoose. Implementing advanced queries, optimizing database performance, and managing complex data relationships in MongoDB.
6	Week 11-12: Operations on data using Node JS	**Competency**: Proficiency in performing data operations and manipulations using Node.js. **Knowledge:** Understanding of data handling techniques, including reading, writing, and processing data with Node.js. **Skills:** Implementing data operations such as file I/O, data transformation, and integration with databases in Node.js applications.

7. Learning Objectives

S/L	**Learning Objectives**	**Description**
1	Module 1: Refreshing JavaScript and CSS	Understand CSS Syntax and Benefits: Learn the basic syntax of CSS and its advantages in web design. Master Responsive Design: Grasp the principles of responsive design and how to implement it using CSS. Get Introduced to Bootstrap: Gain a basic understanding of Bootstrap and its role in building responsive websites. Refresh JavaScript Syntax: Revisit JavaScript fundamentals, including syntax and built-in objects. Handle Errors and Events in JavaScript: Learn how to manage errors and events effectively in JavaScript. Manipulate the DOM: Understand how to use JavaScript for DOM manipulation. Explore Asynchronous Programming: Get introduced to asynchronous programming concepts in JavaScript.
2	Module 2: Introduction to AngularJS	Learn AngularJS Basics and Syntax: Understand the foundational syntax and features of AngularJS. Explore AngularJS Features and Advantages: Identify key features and benefits of using AngularJS. Understand AngularJS Application Structure: Learn how to structure applications using AngularJS. Grasp Routes and Navigation Basics: Get introduced to routing and navigation in AngularJS. Apply MVC Architecture in AngularJS: Learn how to implement MVC architecture using AngularJS. Work with AngularJS Services, Modules, and Directives: Explore how to create and use services, modules, and directives. Build Forms and Validate Data: Learn how to create and validate forms in AngularJS. Implement Data Binding: Understand and apply data binding in AngularJS. Create a Single-Page Website: Build a single-page application using AngularJS.
3	Module 3: Introduction to Node.js Setup	Set Up the Node.js Environment: Learn how to install and configure the Node.js environment. Use the Node.js Package Manager: Understand the basics of npm and how to manage packages in Node.js. Explore Node.js Features and the Console Object: Get familiar with the features of Node.js and the console object. Understand Callbacks in Node.js: Learn the concept of callbacks and their role in Node.js programming.

4	Module 4: Node.js in Detail	Master Events and the Event Loop: Understand the event-driven architecture and the event loop in Node.js. Work with Timers and Error Handling: Learn how to use timers and handle errors in Node.js applications. Handle Buffers and Streams: Explore working with buffers and streams in Node.js. Interact with the File System: Learn how to work with the file system in Node.js. Network with Node.js: Understand TCP, UDP, and HTTP networking in Node.js. Explore Web Modules and Debugging: Learn about web modules and debugging techniques in Node.js. Build REST APIs with Node.js: Create RESTful APIs using Node.js. Manage Sessions and Cookies: Understand how to handle sessions and cookies in Node.js. Implement Design Patterns, Caching, and Scalability: Explore design patterns and strategies for caching and scalability in Node.js applications.
5	Module 5: Database Programming with Node.js and MongoDB	Understand MongoDB Basics and Data Types: Learn the foundational concepts and data types in MongoDB. Connect Node.js to MongoDB: Learn how to establish a connection between Node.js and MongoDB. Perform CRUD Operations: Master data operations like Insert, Find, Query, Sort, Delete, and Update using Node.js with MongoDB.

8. 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
	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

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

Course Outcomes (COs)

COs	Description
M23BCS603C.1	Articulatethe concepts of client side programming using CSS and Java Script
M23BCS603C.2	Apply the concepts of Angular JS to extend basic HTML features
M23BCS603C.3	Analyze Node JS framework to build dynamic server side applications
M23BCS603C.4	Interpretthe concept of database using Mongo DB and connect database with application.

M23BCS603C.5	Design and implement full featured web application using the concepts of Angular JS and Node JS

CO-PO-PSO Mapping

COs/POs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
M23BCS603C.1	**3**												**2**	
M23BCS603C.2		**3**											**2**	
M23BCS603C.3			**3**											**2**
M23BCS603C.4				**3**	**3**									**2**
M23BCS603C.5	**3**	**3**	**3**	**3**	**3**								**2**	**2**
M23BCS603C	**3**	**3**	**3**	**3**	**3**								**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

Pursuing a course in Advanced Web Programming provides a comprehensive foundation in full-stack web development, focusing on both front-end and back-end technologies. Mastering these skills opens up various career opportunities and paths for growth in the tech industry:

1. Full-Stack Developer:

- Versatility: Gain proficiency in both front-end (HTML, CSS, JavaScript, AngularJS) and back-end (Node.js, MongoDB) technologies, making you a versatile developer capable of handling entire projects from conception to deployment. In-Demand Skill Set: Full-stack developers are highly sought after due to their ability to work on various aspects of a web application, leading to numerous job opportunities.

2. Front-End Developer:

- Expertise in User Interfaces: With a strong foundation in CSS, Bootstrap, JavaScript, and AngularJS, you can specialize in creating responsive, dynamic, and user-friendly interfaces.
- Single-Page Applications (SPAs): AngularJS knowledge enables you to build SPAs, which are in high demand for their speed and efficiency.

3. Back-End Developer:

- Server-Side Expertise: Mastery of Node.js equips you with the skills to develop robust, scalable server-side applications.
- API Development: Learn to design and implement RESTful APIs, a crucial component of modern web applications.
- Database Management: With knowledge of MongoDB, you can handle data storage and retrieval efficiently, an essential skill for back-end roles.

4. DevOps Engineer:

- Scalability and Deployment: Understanding Node.js scalability and design patterns can lead to roles focusing on optimizing application performance, deploying web apps, and managing server infrastructure.
- Continuous Integration/Continuous Deployment (CI/CD): Skills in managing sessions, caching, and debugging contribute to a smoother development and deployment pipeline.

5. Specialized Roles:

- JavaScript/Angular Developer: Focus on front-end frameworks and become a specialist in JavaScript and AngularJS, contributing to large-scale projects or working in a niche area of front-end development.
- Node.js Developer: Specialize in Node.js for server-side applications, particularly in building real-time, data-intensive applications like chat apps or collaborative platforms.

6. Entrepreneur/Startup Founder:

- Building MVPs: With the full-stack knowledge, you can create Minimum Viable Products (MVPs) for startups, testing and validating ideas quickly.
- Innovation: The ability to independently build and scale applications allows for innovation and the creation of new products and services.

7. Future Technologies and Trends:

- Adoption of Modern Frameworks: As the industry evolves, the skills acquired here provide a solid foundation for learning and adopting new frameworks and technologies (e.g., Angular's newer versions, MERN stack, microservices architecture).
- Contribution to Open Source: Proficiency in these technologies can lead to contributions to open-source projects, furthering career prospects and community recognition.

8. Continuous Learning and Growth:

- Lifelong Learning: The web development field is constantly evolving. This foundation will make it easier to learn new languages, frameworks, and tools that emerge in the future.

Networking and Collaboration: Understanding these technologies allows you to collaborate effectively with other developers and professionals, leading to networking opportunities and career advancement

6thSemester	Professional Elective -II (PE) SOCIAL NETWORK ANALYSIS	M23BCS603D

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Mathematics	Graph Theory: Understanding concepts like nodes, edges, paths, and centrality measures (e.g., degree centrality, closeness centrality). Probability: Basics of probability theory can be useful for understanding stochastic processes in networks.
2	Statistical	Proficiency in statistical methods to analyse network data, including descriptive statistics, hypothesis testing, and regression analysis.
3	Computational	Programming: Familiarity with programming languages such as Python, R, or Java is important for implementing SNA algorithms and processing data.
4	Data Handling and Management	Data Collection: Skills in collecting and cleaning data from various sources, including social medi, surveys, or databases.
5	Structural Analysis:	Basic Concepts: Understanding fundamental graph theory concepts such as types of networks (e.g., directed, undirected, weighted) and network metrics (e.g., clustering coefficient, network density). Advanced Metrics: Knowledge of more advanced metrics like community detection, network robustness, and dynamic network analysis.
6	Data Interpretation:	Visualization: Ability to create and interpret network visualizations to identify patterns, clusters, and outliers. Statistical Analysis: Skills in applying statistical techniques to validate network findings and draw meaningful conclusions.
7	Practical Application	Case Studies: Experience with real-world data and case studies to understand practical challenges and solutions in network analysis.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction**	**Knowledge:** Semantic Web principles and technologies, Social Web concepts and platforms, Network analysis methodologies Key concepts in social network analysis. **Skills:** Web development basics, Data mining and analysis Use of network analysis tools **Attitudes:** Attention to detail in calculations and diagrams. Persistence in analyzing complex load scenarios.
2	**Modeling and visualization**	**Knowledge:** Graph theory fundamentals Social network analysis principles, Data visualization techniques Centrality measures in network analysis **Skills:** Implementing graph algorithms Creating and interpreting node-edge diagrams Developing matrix-based network representations, Designing hybrid network visualizations **Attitudes:** Analytical mind set for interpreting complex network structures Curiosity about social dynamics and their digital representations
3	**Extraction and mining communities in web social networks**	**Knowledge:** Definitions of social network structures and dynamics Familiarity with community detection algorithms Knowledge of web archiving techniques and formats Understanding of multi-relational and dynamic network models Awareness of decentralized online social networks Grasp of evaluation metrics for community detection.

4		**Skills:** Proficiency in programming languages (e.g., Python, R) Experience with network analysis tools (e.g., Network, Gephi) Ability to implement and adapt community detection algorithms Data pre-processing and cleaning **Attitudes:** Respect for user privacy and data protection. Ethical consideration in behaviour prediction and analysis Commitment to transparency in data collection and use Proactive approach to security and trust issues Curiosity about human behaviour and social dynamics Openness to interdisciplinary approaches
4	**Predicting human behaviour and privacy issues**	**Knowledge:** Data management principles and techniques Statistical inference and distribution methods Reality mining concepts and applications Context-awareness in social computing Privacy concepts and regulations in online environments **Skills:** Data collection and analysis Predictive modelling of human behaviour Designing privacy-preserving systems Implementing trust models and algorithms **Attitudes:** Trust in online environment – Trust models based on subjective logic – Trust network analysis – Trust transitivity analysis – Combining trust and reputation – Trust derivation based on trust comparisons – Attack spectrum and counter measures.
5	**Application**	**Knowledge:** Machine learning algorithms and techniques Linguistic analysis methods , Emotion classification models Social media data structures and APIs , Emerging technology trends **Skills:** Programming (e.g., Python, R) Statistical analysis and Data visualization Model development and evaluation, API integration and Network graph analysis. **Attitudes:** It is use for Commitment to data security.

3. Syllabus

SOCIAL NETWORK ANALYSIS **SEMESTER – VI**			
Course Code	**M23BCS603D**	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: 1. Ability to understand the fundamental concepts and applications of social network analysis 2. Ability to model and visualize the social network 3. Ability to extract and Mine Communities in Web Social Networks 4. Ability to evaluate link prediction techniques and cascade models to predict network behavior and changes. 5. Ability to analyze anomaly detection techniques to identify and address malicious activities in networks.			
Module -1			
Networks and Society , What is Social Network Analysis?, Why do We Study Social Networks? Applications of Social Network Analysis, Preliminaries, Three Levels of Social Network Analysis ,Graph Visualization Tools, Network Measures, Network Basics, Node Centrality, Assortativity ,Transitivity and Reciprocity, Similarity, Degeneracy Text Book1:Chapter 1 ,2			
Module -2			
Network Growth Models, Properties of Real-World Networks, Random Network Model, Ring Lattice Network Model, Watts–Strogatz Model, Preferential Attachment Model, Price’s Model, Local-world Network Growth Model, Network Model with Accelerating Growth, Aging in Preferential Attachment, Link Analysis,			

Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis Algorithms, PageRank,Personalised PageRank, DivRank, SimRank, PathSIM
Text Book1:Chapter 3 ,4

Module -3

Community Structure in Networks, Applications of Community Detection,Types of Communities, Community Detection Methods, Disjoint Community Detection, Overlapping Community Detection,Local Community Detection, Community Detection vs Community Search,Evaluation of Community Detection Methods, Link Prediction, Applications of Link Prediction, Temporal Changes in a Network, Problem Definition
Text Book1:Chapter 5 ,6.1,6.2,6.3

Module -4

Evaluating Link Prediction Methods,Heuristic Models, Probabilistic Models, Supervised Random Walk, Information-theoretic Model, Latest Trends in Link Prediction,Cascade Behaviours and Network Effects ,Preliminaries and Important Terminologies, Cascade Models,Case Study – The "Indignados" Movement , Probabilistic Cascades, Epidemic Models Independent Cascade Models, Cascade Prediction
Text Book1:Chapter 6.4,6.5,6.6,6.7,6.8,6.9, Chapter7

Module -5

Anomaly Detection in Networks,Outliers versus Network-based Anomalies, Challenges,Anomaly Detection in Static Networks, Anomaly Detection in Dynamic Networks Malicious Activities on OSNs, Sockpuppets in OSNs, Collusion on Online Social Networks
Text Book1:Chapter 8,10.1,10.2,10.3

TEXTBOOKS:

1. Social Network Analysis, k Tanmoy Chakraborty , Publisher. Wiley · Publication date. 1 October 2021
2. Guandong Xu ,Yanchun Zhang and Lin Li,-Web Mining and Social Networking – Techniques and applications, First Edition, Springer, 2011.

REFERENCE BOOKS:

1. Dion Goh and Schubert Foo,-Social information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively, IGI Global Snippet, 2008.
2. Max Chevalier, Christine Julien and Chantal Soulé-Dupuy, Collaborative and Social Information Retrieval and Access: Techniques for Improved user Modelling, IGI Global Snippet, 2009.
3. John G. Breslin, Alexander Passant and Stefan Decker, -The Social Semantic Web, Springer, 2009.

VIDEO LINKS:

https://youtu.be/v3JaWbAdTTg
https://youtu.be/hlAwvj60MI4

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: INTRODUCTION NETWORKS AND SOCIETY	Networks and Society, Three Levels of Social Network Analysis ,Graph Visualization Tools, Network Measures, Network Basics, Node Centrality, Assortativity ,Transitivity and Reciprocity, Similarity, Degeneracy
2	Week 4-6: NETWORK GROWTH MODELS AND LINK ANALYSIS	, Properties of Real-World Networks, Random Network Model, Ring Lattice Network Model, Watts–Strogatz Model, Preferential Attachment Model, Price's Model, Local-world Network Growth Model,
3	Week 8-11: COMMUNITY STRUCTURE IN NETWORKS	Community Detection Methods, Disjoint Community Detection, Overlapping Community Detection,Local Community Detection, Community Detection vs Community Search,Evaluation of Community Detection Methods, Link Prediction, Applications of Link Prediction, Temporal Changes in a Network, Problem Definition
4	Week 7-8: EVALUATING LINK PREDICTION METHODS	Heuristic Models, Probabilistic Models, Supervised Random Walk, Information-theoretic Model, Latest Trends in Link Prediction,Cascade Behaviours and Network Effects ,Preliminaries and Important Terminologies, Cascade Models,Case Study – The "Indignados" Movement , Probabilistic Cascades, Epidemic Models Independent Cascade Models, Cascade Prediction

5	Week 9-12: ANOMALY DETECTION IN NETWORKS	Outliers versus Network-based Anomalies, Challenges,Anomaly Detection in Static Networks, Anomaly Detection in Dynamic Networks Malicious Activities on OSNs, Sockpuppets in OSNs, Collusion on Online Social Networks

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	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
5	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
6	Laboratory Learning	Knowledge about tools related to social networks and implementation of social network visualizations using tools such as Gephi, Cytoscape.

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

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	Introduction	Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Statistical Properties of Social Networks Analysis.
2	Modeling and visualization	Visualizing Online Social Networks - A Taxonomy of Visualizations - Graph Representation - Centrality- Clustering - Node-Edge Diagrams - Visualizing Social Networks with Matrix- Based Representations- Node-Link Diagrams
3	Extraction and mining communities in web social networks	Definition of community – Evaluating communities – Methods for community detection and mining – Applications of community mining algorithms – Tools for detecting communities social network infrastructures and communities
4	Predicting human behavior and privacy issues	Understanding and predicting human behaviour for social communities – User data management – Inference and Distribution – Enabling new human experiences – Reality mining – Context – Awareness – Privacy in online social networks – Trust in online environment

5	Application	A Learning Based Approach for Real Time Emotion Classification of Tweets, A New Linguistic Approach to Assess the Opinion of Users in Social Network Environments,

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

Course Outcomes (COs)

Cos	Description
M23BCS603D.1	**Apply** social network analysis techniques, including node centrality, assortativity, transitivity, and reciprocity, to **analyse and interpret** real-world networks.
M23BCS603D.2	***Identify different*** network growth models, link analysis algorithms, and community detection methods to **analyze** and **solve** real-world problems involving network structures, such as social networks, biological networks, and organizational systems
M23BCS603D.3	**Analyse** link prediction methods and network effects, to **interpret** and **examine** the underlying patterns and dynamics in complex networks
M23BCS603D.4	**Analyse** and **evaluate** anomaly detection techniques for static and dynamic networks, including challenges related to malicious activities and collusion in online social networks.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS603D.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS603D.2	3		-	-	2	-	-	-	-	-	-	-	3	-
M23BCS603D.3	-	3	-	-	-	-	-	-	-	-	-	-	-	2
M23BCS603D.4	-	3	-	-	2	-	-	-	-	-	-	-	-	2
M23BCS603D	**3**	**3**	-	-	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	**CO2**	**CO3**	**CO4**	**CO4**	**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**	**CO4**	**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**

10. Future with this Subject:

- ❖ **Integration with Artificial Intelligence (AI):** AI and machine learning will continue to enhance SNA by providing more sophisticated tools for pattern recognition, predictive analytics, and automated insights. AI algorithms can identify trends, anomalies, and influence patterns that may not be apparent through traditional methods.
- ❖ **Big Data and Real-Time Analysis:** With the growth of big data, SNA will increasingly leverage vast amounts of real-time data from social media platforms, communication networks, and other sources. This will enable more dynamic and timely analysis of social interactions and network structures.
- ❖ **Enhanced Visualization Tools:** Advanced visualization techniques will allow for more intuitive and interactive representations of complex social networks. Tools that can effectively illustrate multi-

dimensional relationships and evolving networks will improve our ability to understand and communicate findings.

- **Integration with Other Disciplines:** SNA will benefit from integration with fields like psychology, sociology, and economics. Understanding social behaviour and dynamics in a more holistic context will improve the accuracy and relevance of network analyses.
- **Privacy and Ethical Considerations:** As SNA tools become more powerful, there will be increasing focus on privacy and ethical concerns. Developing frameworks to ensure responsible data use, protect individuals' privacy, and address biases will be critical.
- **Personalized Social Network Insights:** Advances in data analysis will enable more personalized insights into individual behaviours and relationships within networks. This could impact areas like targeted marketing, personalized recommendations, and tailored interventions.
- **Block chain and Decentralized Networks:** Block chain technology and decentralized networks may offer new ways to analyze and visualize social interactions. These technologies could provide greater transparency and security in network analysis.
- **Cross-Network Analysis:** Future SNA will likely involve the analysis of multiple interconnected networks, such as combining social networks with professional or academic networks. This can provide a more comprehensive understanding of individuals' roles and influence across different contexts.
- **Increased Focus on Small-Scale Networks:** While much of SNA has focused on large-scale networks, there will be growing interest in understanding smaller, niche networks, including those within organizations or specific communities.

Advancements in Data Collection Methods: New methods for data collection, such as sensors, wearables, and automated scraping tools, will provide richer and more diverse datasets for analysis, enhancing the depth and accuracy of social network insights.

6th Semester	**Open elective -I** **Programming in Java**	**M23BCS604A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming Concepts**	Understanding the use of variables to store data and understand basic data types like integers, floats, characters, and strings. The use of control structures and Understand how to define and call functions or methods, including parameters and return values.
2	**Understanding of Object-Oriented Programming (OOP)**	Understanding to define classes and create objects in Java and how inheritance allows one class to inherit fields and methods from another. Getting familiar with method overriding and overloading. Learning how to use access modifiers to protect data and ensure a class's internal representation is hidden from the outside.
3	**Basic Knowledge of Java Syntax and Semantics**	Understanding the Java-specific syntax and basic error handling using exceptions.
4	**Development Tools and Environment**	Choosing an IDE or text editor suitable for Java development, such as IntelliJ IDEA, Eclipse, or NetBeans. Learning the usage of the IDE for coding, debugging, and managing your projects. The JDK provides the necessary tools for compiling and running Java programs.
5	**Understanding of Compilation and Execution**	Understanding the difference between source code and bytecode. Understand the process of compiling Java code into bytecode and running it on the Java Virtual Machine (JVM).

2. Competencies

S/L	Competency	KSA Description
1	**Core Java Concepts**	**Knowledge:** Proficiency with Java syntax, including how to write correct and efficient code using Java's syntax rules. **Skills:** Learning the basic concepts of primitive data types in java. **Attitude:** Fundamental to understanding the language and its capabilities.
2	**Object-Oriented Programming**	**Knowledge:** Ability to define and instantiate classes and objects. Understanding method overloading and overriding, and how polymorphism enables flexible code **Skills:** Practicing the concept of functional programming **Attitude:** Principles that shape how developers approach problem-solving and software design.
3	**Java Virtual Machine (JVM) and Performance**	**Knowledge:** Understanding the JVM's role in executing Java applications and how garbage collection works. **Skills:** Understanding the concept of JVM and Execution process. **Attitude:** Component of the Java ecosystem, and its design embodies specific attitudes toward performance and execution.
4	**Development Tools and Practices**	**Knowledge:** Proficiency with build tools like Maven or Gradle for managing dependencies and building projects. Using an Integrated Development Environment (IDE) like IntelliJ IDEA, Eclipse, or NetBeans effectively. **Skills:**

		Understanding the development of new projects using java. **Attitude:** Commitment to efficiency, quality, collaboration, and continuous improvement.

3. Syllabus

Programming in Java
SEMESTER – VI

Course Code	**M23BCS604A**	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.Learn fundamental features of object oriented language and JAVA
2.Set up Java JDK environment to create, debug and run simple Java programs.
3.Learn object oriented concepts using programming examples.
4.Study the concepts of importing of packages and exception handling mechanism.
5.Discuss the String Handling examples with Object Oriented concept

Module -1

An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries.
Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays,
Textbook 1:Ch 2,Ch 3.

Module -2

Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses,
Control Statements: Java’s Selection Statements, Iteration Statements, Jump Statements.
Textbook 1:Ch 4,Ch 5.

Module -3

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method.
A Closer Look at Methods and Classes: Overloading Methods,A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control,Introducing final.
Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding.
Textbook 1: Ch 6, Ch 7.1-7.9, Ch 8.1-8.5

Module -4

Packages and Interfaces:
Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java‟s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.

Module -5

Type Wrappers, I/O, Applets, and Other Topics: I/O Basics, Reading Console Input, Writing Console Output, The PrintWriter Class, Reading and Writing Files, Applet Fundamentals-Example Program.
String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueOf( ), Changing the Case of Characters Within a String , Additional String Methods, StringBuffer, StringBuilder.

Text Books:
1. Herbert Schildt, Java the Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 2, 3, 4, 5, 6,7, 8, 9,10, 12,13,15)
2. Cay S Horstmann, "Core Java - Vol. 1 Fundamentals", Pearson Education, 10th Edition, 2016.

Reference Books:
1. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, "Java 8 in Action", Dreamtech Press/Manning Press, 1st Edition, 2014.

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

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week1-2: **An Overview of Java**	It involves understanding its fundamental aspects, core features. The knowledge of basic concepts such as variables, datatypes, programming structure and execution environment. Acquiring the knowledge of basic concepts of Java programming**.**
2	Week 3-4: **Operators**	Understanding the usage and applying various types of operators effectively. It involves understanding and applying different types of operators. Usage of different operators to perform specific operations.
3	Week 5-6: **Introducing Classes**	Understanding the design, implement, and utilize classes effectively. Encompasses the understanding of classes concept in programming. Ability to use the concept of classes and object to solve related problems in Java.
4	Week 7-8: **Packages and Interfaces**	It involves understanding purpose, usage, and interaction of packages and interfaces within Java applications. Knowledge of packages and interfaces. Involve a range of abilities related to designing, implementing java application using packages and interfaces.
5	Week 9-10: **Type Wrappers, I/O, Applets, and Other Topics**	Involves understanding and effectively using basic concepts of Applets and string methods. Understanding the Applet viewer and different string methods to perform specific operations. Involve effectively using Applets to build basic GUI application.

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	**Pair Programming**	Incorporate pair programming sessions where students collaborate in pairs to solve coding tasks or work on projects together.
7	**Use of Tools and Resources**	Familiarize students with IDEs like IntelliJ IDEA or Eclipse for coding and debugging. Apply Java concepts to practical problems and projects to demonstrate their utility.
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)

Theory Course with 3 Credits: Open Elective (OE)

This section of regulations is applicable to all theory-based courses. The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Open Elective (OE)

	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 Examinations:

Theory Course with 4, 3 and 2 Credits: Professional Core Course (PC)/Professional Elective/Open Elective

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	**Master Advanced Java Concepts**	Ensuring that students gain a deep understanding of complex Java concepts, tools, and frameworks, enabling them to develop robust, scalable, and efficient applications
2	**Apply Object-Oriented Programming principles**	Creating and manipulating classes and objects, and understand the role of constructors. Implementing and using inheritance to create hierarchical relationships between classes. Utilizing method overloading and overriding to achieve polymorphism.
3	**Understand Java's Execution Environment**	Java code is compiled into bytecode and executed on the Java Virtual Machine (JVM). Understand basic concepts of memory management, including garbage collection.
4	**Practical Application**	Developing Practical Programming Skills. Designing and implementing the algorithms for common problems and tasks

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

Course Outcomes (COs)

COs	Description
M23BCS604A.1	Understanding the basic concepts of java to solve real time problems.
M23BCS604A.2	Apply the object-oriented concepts of java and exception handling concepts to implement java program.
M23BCS604A.3	Analyze I/O and String handling concept to develop an application program.
M23BCS604A.4	Analyze and develop computer programs to solve real world problems in Java.

CO-PO-PSO Mapping

COs/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2
M23BCS604A.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-
M23BCS604A.2	-	3	-	-	-	-	-	-	-	-	-	-	-	-
M23BCS604A.3	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCS604A.4	-	-	-	3	-	-	-	-	-	-	-	3	3	3
M23BCS604A	3	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**	**20**		**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	40		**50**

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 advanced Java development is poised to evolve with advancements in technology and shifts in software development practices.

1. Language Evolution: Ongoing updates and feature enhancements in recent versions improve performance, simplify syntax, and introduce new functionalities. This aims to simplify concurrency and scalability with lightweight, user-friendly fibers and continuations.

2. Performance Improvements: The Java Virtual Machine (JVM) continues to receive performance optimizations, improving execution speed and efficiency.

3. **Integration with Emerging Technologies:** Java is increasingly used in big data processing frameworks (like Apache Hadoop and Apache Spark) and artificial intelligence applications. Java's portability and robustness make it suitable for IoT applications and devices.

4. Educational and Enterprise Adoption: Java remains a primary language taught in computer science programs, ensuring a steady influx of new developers. Many large enterprises continue to rely on Java for mission-critical applications, ensuring its relevance in the business world.

5. Development Practices and Tools: Improved tools and practices for monitoring and managing the performance and health of Java applications in production environments. Advanced Integrated Development Environments (IDEs) and tools providing enhanced code assistance, debugging, and profiling capabilities.

6th Semester	**Open Elective** **Introduction to Data Structures**	**M23BCS604B**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Programming Concepts**	**Variables and Data Types:** Understanding different data types (int, float, char, etc.) and how to declare and use variables. **Control Structures:** Proficiency with loops (for, while, do-while) and conditional statements (if, else if, else, switch). **Functions:** Understanding how to define and use functions, including parameter passing, return types, and recursion. **Pointers:** Understanding pointers, pointer arithmetic, and how they relate to arrays and functions. **Memory Management:** Familiarity with dynamic memory allocation using malloc (), calloc (), realloc (), and free () functions.
2	**Advanced C Programming Concepts**	**Structures:** Ability to define and use structures to group different data types together. **Unions and Enumerations:** Understanding how to use unions and enumerations. **File I/O:** Basic knowledge of reading from and writing to files in C. **Preprocessor Directives:** Familiarity with macros, #define, #include, and conditional compilation.
3	**Basic Algorithms**	**Sorting Algorithms:** Understanding basic sorting algorithms like Bubble Sort, Selection Sort, and Insertion Sort. **Searching Algorithms:** Familiarity with searching techniques like Linear Search and Binary Search.
4	**Mathematical Concepts**	**Discrete Mathematics:** Basic understanding of sets, functions, relations, and combinatorics.
5	**Problem-Solving Skills**	**Algorithm Design:** Ability to design algorithms for solving problems using data structures. **Debugging and Testing:** Skills in debugging code and testing to ensure correctness and efficiency.

2. Competencies

S/L	Competency	KSA Description
1	**In-Depth Understanding of Data Structures**	**Knowledge (K)** **Arrays:** Understanding static and dynamic arrays, multidimensional arrays, and their applications. **Linked Lists:** Knowledge of singly linked lists, doubly linked lists, and circular linked lists, including their memory structure and use cases. **Stacks and Queues:** Understanding of stack and queue concepts, including their implementation and applications using arrays and linked lists. **Skills (S)** **Coding Proficiency:** Ability to write, debug, and optimize code for implementing various data structures from scratch in C. **Data Structure Operations:** Skill in implementing operations such as insertion, deletion, searching, and traversal for different data structures. **Attitudes: Use of Pointers and Dynamic Memory:** Proficiency in using pointers for creating and manipulating complex data structures, and managing dynamic memory effectively.
2	**Problem-Solving Abilities**	**Knowledge (K)** **Complexity Analysis:** Knowledge of time and space complexity analysis using Big O notation, with the ability to analyze the performance of data structure operations.

		Sorting and Searching Algorithms: Understanding of fundamental algorithms and their integration with data structures. **Recursion and Iteration:** Knowledge of recursive and iterative approaches to solving problems, especially in tree and graph algorithms. **Skills (S)** **Data Structure Selection:** Ability to choose the most appropriate data structure for solving specific problems based on efficiency and complexity considerations. **Debugging:** Skill in identifying and fixing bugs related to pointers, memory allocation, and data structure operations in C code. **Optimization:** Ability to optimize data structures for performance, minimizing time and space complexity through efficient algorithms and code practices. **Attitudes: Practical Application:** Ability to apply knowledge of data structures in solving real-world programming challenges, particularly in system programming, embedded systems, and performance-critical applications. **Scalability and Efficiency:** Ability to design and implement data structures that scale efficiently with large datasets or under high-performance requirements
3	**Errors and Handling the Errors**	**Knowledge:** Understanding of issues with errors. **Skills:** Implementing how to handle the errors through appropriate C++ programming construct. **Attitudes: Appreciation for the way error is handled and making the execution of program in control.**
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.

3. Syllabus

Introduction to Data Structures **SEMESTER – VI**			
Course Code	**M23BCS604B**	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 1. Introduce elementary data structures. 2. Analyze Linear Data Structures: Stack, Queues, Lists 3. Analyze Non Linear Data Structures: Trees 4. Assess appropriate data structure during program development/Problem Solving.			
Module -1			
Introduction: Introduction to arrays: one-dimensional arrays, two dimensional arrays, initializing two dimensional arrays, Multidimensional arrays. Introduction to Pointers: Pointer concepts, accessing variables through pointers, Dynamic memory allocation, pointers applications. Introduction to structures and unions: Declaring structures, Giving values to members, structure initialization, arrays of structures, nested structure, unions, sizeof() structures. **Textbook 1: Ch 8.3 to 8.15,Ch 12.3 to 12.19** **Textbook 2:Ch 2.1 to2.13,2.51 ,2.80 to 2.98**			
Module -2			
Linear Data Structures-Stacks and queues: Introduction, Stack representation in Memory, Stack Operations, Stack Implementation, Applications of Stack. Introduction, Queues-Basic concept, Logical representation of Queues, Queue Operations and its types, Queue Implementation, Applications of Queue. **Textbook 2: Ch 6.1 to 6.14 ,Ch 8.1,8.2**			

Module -3

Linear Data Structures-Linked List:
Introduction, Linked list Basic concept, Logical representation of Linked list, Self-Referential structure, Singly-linked List Operations and Implementation, Circular Linked List, applications of Linked list.
Textbook 1: Ch 15.1 ,15.3,15.4,15.8
Textbook 2: Ch 9.2.9.5

Module -4

Non Linear Data Structures – Trees
Introduction, Basic concept, Binary Tree and its types, Binary Tree Representation, Binary Tree Traversal, and Binary Search tree, Expression Trees.
Textbook1: Ch 16.1,16.2
Textbook2:Ch 10.1,10.2,10.4,10.6.3

Module -5

Sorting and Searching
Sorting: Introduction, Bubble sort, Selection sort, Insertion sort
Searching: Introduction, Linear search, Binary search.
Textbook1: Ch 17.1,17.2.2, 17.2.4, 17.3.1,17.3.2
Textbook2: Ch 11.1.,11.2,11.3,11.7,11.10.1,11.10.2

Textbooks

1. C Programming and data structures, E Balaguruswamy 4th Edition, 2007, McGraw Hill
2. Systematic approach to Data structures using C, A M Padma Reddy, 7thEdition 2007, Sri Nandi Publications.

References

1. Ellis Horowitz and Sartaj Sahni, 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.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week1-2: **Introduction to arrays, Introduction to Pointers**	Introduction to arrays: one-dimensional arrays, two dimensional arrays, initializing two dimensional arrays, Multidimensional arrays. Introduction to Pointers: Pointer concepts, accessing variables through pointers, Dynamic memory allocation, pointers applications. Introduction to structures and unions: Declaring structures, Giving values to members, structure initialization, arrays of structures, nested structure, unions, sizeof() structures.
2	Week 3-4: **Stack and Queues**	Introduction, Stack representation in Memory, Stack Operations, Stack Implementation, Applications of Stack. Introduction, Queues-Basic concept, Logical representation of Queues, Queue Operations and its types, Queue Implementation, Applications of Queue.
3	Week 5-6: **Linear Data Structures-Linked List:**	Introduction, Linked list Basic concept, Logical representation of Linked list, Self-Referential structure, Singly-linked List Operations and Implementation, Circular Linked List, applications of Linked list.
4	Week 7-8: **Non Linear Data Structures – Trees**	Introduction, Basic concept, Binary Tree and its types, Binary Tree Representation, Binary Tree Traversal, Binary Search tree, Expression Trees.
5	Week 9-10: **Sorting and Searching**	Sorting: Introduction, Bubble sort, Selection sort, Insertion sort Searching: Introduction, Linear search, Binary search.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description

1	**Foundation in C Programming**	**Prerequisite Knowledge**: Ensure students have a solid understanding of C programming basics, including pointers, memory allocation, and structures (struct). **Review Pointers and Memory Management**: Since data structures in C heavily rely on pointers, start with a review of pointer operations and dynamic memory allocation using malloc(), calloc(), realloc(), and free().
2	**Begin with Basic Data Structures**	**Arrays and Strings**: Start with arrays as the simplest form of data structure in C. Teach how strings are handled as arrays of characters, and introduce basic string manipulation functions. **Structures (struct)**: Introduce the concept of structures to group different data types, laying the groundwork for more complex data structures like linked lists and trees.
3	**Introduce Linked Lists**	**Singly Linked List**: Begin with singly linked lists, covering concepts like nodes, head pointers, and traversal. Emphasize how pointers are used to link nodes. **Implementation**: Guide students through coding linked list operations such as insertion, deletion, and traversal. **Debugging Practice**: Since pointer errors are common, provide debugging exercises to help students develop problem-solving skills.
4	**Stack and Queue Implementations**	**Stack Using Arrays and Linked Lists**: Teach how to implement stacks using both arrays and linked lists. Discuss the advantages and disadvantages of each approach. **Queue Implementation**: Similar to stacks, introduce queues with both array-based and linked-list-based implementations. Explain the differences between simple queues, circular queues, and priority queues.
5	**Dynamic Memory Management**	**Memory Allocation for Data Structures**: Use examples to show how to allocate and deallocate memory for data structures dynamically. Discuss memory leaks and best practices for managing memory in C.
6	**Advanced Data Structures**	**Trees**: Introduce binary trees and binary search trees, focusing on recursive implementations of tree operations. Explain traversal methods (in-order, pre-order, post-order) and their applications. **Graphs**: Teach basic graph representations using adjacency matrices and adjacency lists. Discuss graph traversal algorithms (BFS and DFS) and their implementation in C. **Hash Tables**: Explain the concept of hashing and collision resolution techniques. Implement a simple hash table using arrays and linked lists.

6. Assessment Details (both CIE and SEE)

Theory Course with 3 Credits: Open Elective (OE)

This section of regulations is applicable to all theory-based courses. The minimum CIE marks requirement is 40% of maximum marks in each component.

CIE Split up for Open Elective (OE)

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 Examinations:

Theory Course with 4, 3 and 2 Credits: Professional Core Course (PC)/Professional Elective/Open Elective

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 Concepts of Data Structures**	**Definition and Types**: Understand what a data structure is and the different types (e.g., linear and non-linear). **Efficiency**: Learn the importance of data structures in terms of time and space complexity. **Data Structure Operations**: Master the basic operations (insertion, deletion, traversal, searching, and sorting) on various data structures.
2	**Mastering Linear Data Structures**	**Arrays**: Learn how to use arrays, including dynamic arrays, and understand their memory management. **Linked Lists**: Understand the implementation of singly linked lists, doubly linked lists, and circular linked lists. **Stacks**: Study stack operations (push, pop, peek) and their implementation using arrays or linked lists. **Queues**: Understand queues, including circular queues, and their implementation using arrays or linked lists.
3	**Understanding Non-Linear Data Structures**	**Trees**: Learn about binary trees, binary search trees (BST), AVL trees, and heap trees. Understand tree traversal methods (in-order, pre-order, post-order). **Graphs**: Understand the representation of graphs using adjacency matrices and lists. Study graph traversal techniques (BFS and DFS). **Hash Tables**: Learn about hashing, hash functions, and collision resolution techniques.
4	**Practical Implementation Skills**	**C Programming**: Enhance C programming skills, particularly in relation to implementing data structures from scratch. **Problem Solving**: Apply data structures to solve real-world problems, understanding when and how to use each type of structure. **Debugging and Optimization**: Learn to debug code effectively and optimize data structures for performance.

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

Course Outcomes (COs)

Cos	Description
M23BCS604B.1	Apply basic concepts of data structures and linear data structure to solve computational problems.
M23BCS604B.2	Apply dynamic memory management techniques using pointers and implement complex data structures.
M23BCS604B.3	Analyze non-linear data structures like trees and graphs.
M23BCS604B.4	Analyze various algorithms for sorting, searching, and other data operations.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS604B.1	**3**	-	-	-	**3**	-	-	-	-	-	-	-	3	3
M23BCS604B.2	**3**	**-**	-	-	**3**	-	-	-	-	-	-	-	3	
M23BCS604B.3	-	**3**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**		3
M23BCS604B.4	-	**3**	-	-	**3**	-	-	-	-	-	-	-	3	3
M23BCS604B	**3**	**3**	-	-	**3**	-	-	-	-	-	-	-	**3**	**3**

Explanation:

- **CO1and CO2** primarily maps to **PO1** because it involves applying foundational knowledge to solve problems.

- **CO3** and **CO4** relate **PO2**, as they require students to analyze various data structures, which aligns with problem-solving.
- **CO1** to **CO4** maps to **PO5** because student use modern tools (such as C programming language).

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				10	10
Total	**10**	**10**	**10**	**20**	**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				20	20
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

Foundation for Advanced Computer Science Concepts

- **Algorithms**: Data structures are the building blocks for designing and analyzing algorithms. A solid understanding of data structures in C prepares you for more advanced topics like algorithm design, complexity analysis, and optimization.
- **Systems Programming**: C is often used in system-level programming (e.g., operating systems, embedded systems). Understanding data structures is crucial for writing efficient and high-performance system code.
- **Artificial Intelligence and Machine Learning**: Efficient data management using appropriate structures is key in AI/ML for handling large datasets and implementing algorithms.
- **Career Opportunities**
- **Software Development**: Knowledge of data structures is essential for software engineers, as it directly impacts the efficiency and performance of software applications. Companies like Google, Microsoft, and Amazon highly value this skill.
- **System Architect or Engineer**: In roles that involve designing large-scale systems (e.g., distributed systems, databases), data structures are vital for ensuring scalability and performance.
- **Embedded Systems Engineer**: C is the language of choice for embedded systems, where efficient data structures are crucial due to limited resources.
- **Database Administrator/Developer**: Understanding data structures helps in optimizing database queries, designing indexing strategies, and improving overall database performance.
- **Game Development**: Game developers use data structures extensively for managing game states, rendering graphics, and handling user input in real-time.

6th Semester	Open Elective –I(OE) Introduction to DBMS	M23BCS604C

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

Introduction DBMS **SEMESTER – VI**			
Course Code	**M23BCS604C**	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:

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 demonstrate the use of concurrency and transactions in database.
5. To design and build database application for real world problems.
6. To become familiar with database storage structures and access techniques.

Module -1

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

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 **8hours**

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 hours**

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

Functional dependencies: **8 hours**

Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies,

Text-1: CH-14.1-14.3

Module -5

Normalization:

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-14.4-14.7

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

CIE Split up for Open Elective (OE)

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 Examinations:

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
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
M23BCS604C.1	Understand and apply the basic elements of a relational database management system.
M23BCS604C.2	Apply various constraints, techniques and Structured Query Language (SQL) statement for database operations.
M23BCS604C.3	Analyze various database models and normalization for the given application.
M23BCS604C.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**
M23BCS604C.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCS604C.2	3	-	-	-	2	-	-	-	-	-	-	-	3	3
M23BCS604C.3	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCS604C.4	-	-	3	-	-	-	-	-	-	-	-	-	-	3
M23BCS604C	**3**	**3**	**3**		**2**								**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				**10**	**10**
Total	**10**	**10**	**10**	**20**	**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				**20**	**20**
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 mark

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.,

6th Semester	**Open Elective-I(OE)** **Introduction to Operating system**	**M23BCS604D**

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. **Attitudes**: Ability to 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. **Attitudes**: Ability to 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. **Attitudes**: Ability to 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. **Attitudes**: Ability to 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. **Attitudes**: Ability to implement and manage process scheduling strategies, optimize resource allocation, and troubleshoot process deadlocks or race conditions.

3. Syllabus

Introduction to Operating system
SEMESTER – VI

Course Code	**M23BCS604D**	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. Basic Understanding of Computer System Structure and Operating Systems Structure
2. Analyze the main tasks carried out by the operating systems Process and thread management, CPU scheduling algorithms, memory management and deadlocks.
3. 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)

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

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 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)

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

CIE Split up for Open Elective (OE)

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 Examinations:

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 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.
5	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
M23BCS604D.1	Understand the fundamental concepts and principles of operating systems.
M23BCS604D.2	Analyze various inter-process communication, multiprogramming mechanisms and apply different process scheduling algorithms.
M23BCS604D.3	Examine multiple mechanisms for managing deadlock situations and Implement both software and hardware solutions to address the critical-section problem
M23BCS604D.4	Implement and evaluate memory management techniques.
M23BCS604D.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
M23BCS604D.1	3	-	-	-		-	-	-	-	-	-	3	3	
M23BCS604D.2	-	-	3	2	3	-	-	-	-	-	-	3	3	
M23BCS604D.3	-	-	3	2	3	-	-	-	-	-	-	3		3
M23BCS604D.4	3	-	3	2	3	-	-	-	-	-	-	3	3	
M23BCS604D.5	-	-	3	-		-	-	-	-	-	-	-	3	
M23BCS604D	**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.

10. Future with this Subject

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

- 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.
- 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.
- 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.
- 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.
- Virtualization and Containerization: Virtualization and containerization technologies will continue to play a crucial role in managing and deploying applications. Future operating systems may provide more native support for these technologies, making it easier to create and manage isolated environments for running application.

6thSemester	Project Work (PW) MAJOR PROJECT PHASE-I	M23BCS605

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Understanding Research Methodology	Basic understanding of research methods, gained from prior courses in engineering mathematics and introductory project work.
2	Conducting a Literature Survey	Familiarity with academic databases, journals, and research papers; understanding of the subject matter from core courses.
3	Defining a Problem Statement	Critical thinking and analytical skills, developed through previous coursework in related engineering disciplines.
4	Multidisciplinary Collaboration	Basic knowledge of related disciplines (e.g., Mechanical students should have a basic understanding of Electronics, etc.).
5	Technical Communication	Writing technical reports and presenting technical content

2. Competencies

S/L	Competency	KSA Description
1	Research Skills	**Knowledge:** Understanding of advanced research methods and tools. **Skill:** Ability to identify, review, and synthesize relevant literature. **Attitude:** Commitment to thorough investigation and unbiased analysis.
2	Problem Identification	**Knowledge:** Deep understanding of the chosen topic area. **Skill:** Capability to define and frame a research problem effectively. **Attitude:** Critical and innovative thinking.
3	Technical Writing	**Knowledge:** Familiarity with technical writing conventions. **Skill:** Proficiency in drafting structured, clear, and concise reports. **Attitude:** Attention to detail and accuracy in documentation.
4	Presentation Techniques	**Knowledge:** Understanding of effective communication strategies. **Skill:** Ability to create and deliver engaging presentations. **Attitude:** Confidence and poise in public speaking.

3. Project Timeline

S/L	Timeline	Description
1	**Week 1-2**	Introduction to research methods and tools; exploring literature review techniques.
2	**Week 3-4**	Initiating literature survey; identifying key research papers and sources.
3	**Week 5-6**	Analysis and synthesis of literature; identifying gaps and formulating insights.
4	**Week 7-8**	Defining the problem statement based on literature findings.
5	**Week 9-10**	Drafting the initial report; focusing on structure and content.
6	**Week 11-12**	Finalizing the report and preparing the presentation.
7	**Week 13-14**	Presentation rehearsal; peer review and feedback sessions
8	**Week 15**	Submission of the final report and formal presentation.

4. Course Objectives

- To enable students to conduct a comprehensive literature survey related to their project topic.
- To guide students in defining a clear and feasible problem statement.
- To develop skills in report writing, summarizing findings, and formal presentation.

5. Assessment Details (both CIE and SEE)

CIE procedure for ProjectWork Phase-I:

(1) Single discipline: The CIE marks shall be awarded by a committee consisting of the Head of the concerned Department and two senior faculty members of the
Department, one of whom shall be the Guide.

The CIE marks awarded for the project work shall be based on the evaluation of the project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates.

(2) Interdisciplinary: Continuous Internal Evaluation shall be group-wise at the college level with the participation of all guides of the college. Participation of external guide/s, if any, is desirable. The CIE marks awarded for the project work, shall be based on the evaluation of project work Report, project presentation skill, and question and answer session in the ratio 50:25:25. The marks awarded for the project report shall be the same for all the batch mates.

SEE procedure for Project Work Phase-I: There shall be no SEE.

6. Learning Objectives

S/L	Learning Objectives	Description
1	Understand the process of conducting a literature survey.	Students will gain expertise in identifying and reviewing relevant research literature.
2	To formulate a research problem statement.	Students will learn to define a research problem that is clear, concise, and researchable.
3	To enhance technical writing and presentation skills.	Students will develop the ability to draft detailed reports and present their findings effectively.

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

Course Outcomes (COs)

Cos	Description
M23BCS605.1	Conduct a comprehensive literature survey and synthesize key findings.
M23BCS605.2	Define a research problem statement based on literature review.
M23BCS605.3	Develop and present a well-structured project report.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS605.1	3	3	-	3	-	3	-	-	-	-	-	-	3	3
M23BCS605.2	-	3	3	-	3	-	-	3	-	-	-	-	3	3
M23BCS605.3	3	-	3	3	-	-	3	-	3	3	3	3	3	3
M23BCS605	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

8. Future with this Subject

This phase equips students with essential research and analytical skills, forming the foundation for the practical work in Phase II. It also enhances their technical writing and presentation abilities, which are critical for their final year projects and professional careers.

6th Semester	**Professional Core Laboratory(PCL)** **Machine Learning Lab**	**M23BCSL606**

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Foundational Programming in Python.	• Understanding data types (integers, floats, strings, lists, dictionaries) • Control flow (loops, conditional statements) • Functions, Basic file handling.
2.	Working with Data using Pandas	• Loading data from files (CSV, etc.) • Dataframe manipulation (selecting columns, filtering rows, adding/removing data) • Handling missing values.
3.	Data Visualization Basics	• Creating plots using Matplotlib or Seaborn (histograms, scatter plots, bar charts, etc.) • Customizing plots (labels, titles, legends).
4.	Core Mathematical Concepts	• Basic linear algebra (vectors, matrices, linear equations - at a conceptual level) • Basic probability and statistics (mean, variance, probability distributions, conditional probability).
5.	Introduction to Machine Learning Concepts.	• Understanding of the machine learning process (training data, testing data, model building, evaluation) • Familiarity with the types of machine learning (supervised, unsupervised).

2. Competencies

S/L	Competency	KSA Description
1	Data Fluency and Exploration	**Knowledge:** • Understanding of various data types (numerical, categorical, text) and their characteristics. • Awareness of data quality issues and common data cleaning techniques. • Familiarity with key descriptive statistics and visualization methods. **Skills:** • Ability to load, clean, transform, and visualize data using Python libraries like Pandas and Matplotlib/Seaborn. • Skill in identifying patterns, trends, and potential relationships within datasets. **Attitudes:** • Appreciation for the importance of data quality and its impact on ML model performance. • Curiosity and a data-driven mindset when approaching problems.
2	Supervised Learning Techniques	**Knowledge:** • Understanding of supervised learning principles, including regression and classification. • Familiarity with common algorithms: Linear Regression, Logistic Regression, Decision Trees, Random Forests. • Knowledge of model evaluation metrics (e.g., MSE, accuracy, precision, recall, F1-score). **Skills:** • Ability to build, train, and evaluate supervised learning models using scikit-learn or similar libraries. • Skill in tuning hyper parameters to optimize model performance. • Ability to interpret model results and communicate findings effectively. **Attitudes:** • Critical thinking when selecting and evaluating models, considering bias-variance trade-offs.

3	Unsupervised Learning Techniques	**Knowledge:** • Understanding of clustering and dimensionality reduction techniques. • Familiarity with algorithms like k-Means Clustering and Principal Component Analysis (PCA). **Skills:** • Ability to apply clustering to discover patterns and group similar data points. • Skill in using dimensionality reduction to visualize high-dimensional data and improve model efficiency**.** **Attitudes:** • Openness to exploring data without predefined labels. • Understanding the value of uncovering hidden structures and relationships within data.
4	Reinforcement Learning Foundations	**Knowledge:** • Basic understanding of reinforcement learning concepts (agents, environments, rewards, policies). • Introduction to Q-learning or other fundamental RL algorithms. **Skills:** • Ability to implement a simple Q-learning agent in a simulated environment. **Attitudes:** • Interest in the potential of learning through trial and error.

3. Syllabus

Machine Learning Laboratory
SEMESTER – VI

Course Code	**M23BCSL606**	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:

1. Master fundamental machine learning concepts and algorithms across various paradigms (supervised, unsupervised, reinforcement).
2. Develop proficiency in implementing and evaluating ML models using Python and relevant libraries.
3. Cultivate a data-driven approach to problem-solving through exploration, visualization, and insight generation.
4. Critically evaluate ML model performance, considering factors beyond accuracy and selecting appropriate models for diverse tasks.
5. Promote a spirit of exploration, independent learning, and continuous growth in the ever-evolving field of machine learning.

1: Data Exploration and Visualization

Aim: Familiarize with basic data loading, cleaning, and visualization techniques.

Question: Analyze a dataset of your choice (e.g., Iris dataset, Titanic dataset) to identify patterns and relationships between variables.

Dataset:

Iris: https://www.kaggle.com/datasets/uciml/iris

Titanic: https://www.kaggle.com/c/titanic.

2: Data Preprocessing Pipeline

Aim: Build a data preprocessing pipeline to handle missing values, categorical data, and feature scaling.

Question: Prepare a dataset for a machine learning task. Implement techniques like imputation, one-hot encoding, and standardization/normalization.

Dataset: Use the same dataset from Program 1 or a different one with similar characteristics.

3: Implementing k-Nearest Neighbors

Aim: Understand and implement the k-NN algorithm for classification.

Question: Build a k-NN classifier to predict the species of iris flowers based on sepal and petal measurements. Experiment with different values of 'k'**.**

Dataset: Iris dataset.

4: Linear Regression for Prediction
Aim: Apply linear regression to predict a continuous target variable.
Question: Build a linear regression model to predict housing prices based on features like area, number of bedrooms, and location.
Dataset:
Boston Housing: https://www.kaggle.com/datasets/vikrishnan/boston-house-prices
California Housing**:** https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset

5: Logistic Regression for Classification
Aim: Use logistic regression for binary or multi-class classification.
Question: Build a logistic regression model to predict whether a customer will click on an ad based on demographics and browsing history.
Dataset: You can simulate this type of dataset or find related advertising datasets on Kaggle.

6: Comparing Regression Models
Aim: Compare the performance of different regression models on a given dataset.
Question: Evaluate and compare linear regression, polynomial regression, and Ridge/Lasso regression for a prediction task of your choice.
Dataset: Use a dataset suitable for regression, potentially from previous programs.

7: Decision Tree Classification
Aim: Visualize and interpret decision tree models.
Question: Build a decision tree classifier to predict customer churn based on service usage patterns and account information. Visualize the tree and analyze feature importance.
Dataset: You can simulate a churn dataset or search for "telecom churn" datasets.

8: Ensemble Methods - Random Forest
Aim: Apply the Random Forest algorithm and evaluate its performance.
Question: Build a Random Forest classifier for a classification task. Tune hyperparameters (e.g., number of trees) to optimize performance.
Dataset: Choose a dataset suitable for classification, possibly from a previous program.

9: Naive Bayes for Text Classification
Aim: Implement the Naive Bayes algorithm for text data.
Question: Build a spam email classifier using the Naive Bayes algorithm.
Dataset: Spambase: https://archive.ics.uci.edu/ml/datasets/Spambase
Other text datasets: https://www.kaggle.com/datasets?tags=text

10: Customer Segmentation with K-Means Clustering
Aim: Apply clustering to group similar data points.
Question: Segment customers based on their purchasing behavior (e.g., RFM: Recency, Frequency, Monetary value) using k-means clustering. Visualize the clusters.
Dataset: Simulate a customer purchase dataset or use a retail dataset with transaction history.

11: Dimensionality Reduction with PCA
Aim: Reduce data dimensionality while preserving important information**.**
Question: Apply PCA to a dataset with a high number of features (e.g., images, text) and visualize the data in a lower-dimensional space.
Dataset:
MNIST Handwritten Digits (image): http://yann.lecun.com/exdb/mnist/
20 Newsgroups (text): https://scikit-learn.org/stable/datasets/real_world.html#the-20-newsgroups-text-dataset

12: Introduction to Q-Learning
Aim: Implement a basic reinforcement learning algorithm.
Question: Use Q-learning to train an agent to navigate a simple grid-world environment and find an optimal path to a goal.
Dataset: No external dataset is needed. You'll create a grid environment within your code (e.g., using Python lists).

TEXTBOOK:
Prescribed Textbooks:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron (3rd Edition)
2. Python Machine Learning by Sebastian Raschka and Vahid Mirjalili.

REFERENCE BOOKS:

1. Tom Michel, Machine Learning, McGrawHill Publication.
2. Introduction to Machine Learning, Fourth Edition By Ethem Alpaydin.

VIDEO LINKS:

1.Machine Learning Course by Andrew Ng (Stanford CS229): https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU

2. Data School: https://www.youtube.com/user/dataschool

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week-1** Data Exploration and Visualization	Lecture: Introduction to Machine Learning, Types of Learning, Applications, Python for ML Lab: Program 1 - Data Exploration and Visualization
2	**Week-2** Data Preprocessing	Lecture: Data Preprocessing Techniques (handling missing values, encoding categorical data, feature scaling) Lab: Program 2 - Data Preprocessing Pipeline
3	**Week 3** k-Nearest Neighbours	Lecture: Distance Metrics, k-NN Algorithm, Model Evaluation (accuracy, confusion matrix) Lab: Program 3 - Implementing k-Nearest Neighbors
4	**Week 4** Linear Regression	Lecture: Linear Regression Fundamentals, Cost Functions, Gradient Descent (Conceptual), Simple Linear Regression Lab: Program 4 - Linear Regression for Prediction
5	**Week 5** Multiple Linear Regression & Logistic Regression	Lecture: Multiple Linear Regression, Assumptions of Linear Regression, Feature Selection, Introduction to Classification Lab: Program 5 - Logistic Regression for Classification.
6	**Week 6** Model Evaluation & Comparison (Regression)	Lecture: Overfitting and Underfitting, Bias-Variance Trade-off, Regularization (Ridge, Lasso) Lab: Program 6 - Comparing Regression Models.
7	**Week 7:** Decision Trees	Lecture: Decision Tree Learning, Entropy, Information Gain, Visualizing Decision Trees Lab: Program 7 - Decision Tree Classification
8	**Week 8:** Ensemble Methods	Lecture: Ensemble Learning Concepts (Bagging, Boosting), Random Forests Lab: Program 8 - Ensemble Methods - Random Forest.
9	**Week 9:** Naive Bayes & Text Preprocessing	Lecture: Bayes' Theorem, Naive Bayes Classifier, Text Preprocessing Techniques (Tokenization, Stop Words). Lab: Program 9 - Naive Bayes for Text Classification.
10	**Week 10:** Unsupervised Learning: Clustering	Lecture: Introduction to Unsupervised Learning, Clustering Algorithms, k-Means Clustering Lab: Program 10 - Customer Segmentation with k-Means Clustering.
11	**Week 11:** Dimensionality Reduction	Lecture: Curse of Dimensionality, Feature Extraction, Principal Component Analysis (PCA) Lab: Program 11 - Dimensionality Reduction with PCA
12	**Week 12:** Introduction to Reinforcement Learning	Lecture: Fundamentals of Reinforcement Learning, Q-Learning Lab: Program 12 - Introduction to Q-Learning.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Theory-Practice Bridge	Begin each lab with a brief recap of relevant ML concepts from lectures. Prompt students to connect theoretical understanding with the lab's practical implementation.

2	Scaffolding to Independence	Start with partially completed code templates (scaffolding) for early programs. Gradually reduce scaffolding as labs progress, leading students to write more code independently.
3	Collaborative Learning	Integrate pair programming, group discussions, and peer code reviews. Encourage students to help each other debug, analyze, and explore alternative approaches.
4	Interactive Visualizations	Utilize tools like Jupyter Notebooks to encourage interactive data and model exploration. Guide students to create insightful visualizations that deepen their understanding.
5	Real-World Projects	Group several labs into a larger project using a real-world dataset. This provides context, boosts motivation, and mirrors industry workflows.

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):

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**

1. SEE marks for practical course shall be 50 marks
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	Master fundamental ML concepts and algorithms across paradigms.	This objective sets a strong foundation, ensuring students understand not just how to code, but why algorithms work, covering supervised, unsupervised, and reinforcement learning.
2	Develop proficiency in implementing and evaluating ML models using Python	This focuses on the essential practical skills: coding, using libraries, and assessing model performance. Python is a wise choice given its popularity in the ML community.
3	Cultivate a data-driven approach	This highlights the importance of data exploration, visualization, and insight generation, which are essential for real-world problem-solving. It's not just about building models blindly.
4	Critically evaluate ML model performance.	This promotes a deeper understanding of model selection and evaluation. It moves beyond simple accuracy metrics to consider factors like bias-variance trade-off, interpretability, and suitability for specific tasks.
5	Promote a spirit of exploration	This is crucial in the rapidly evolving field of ML. It encourages students to be lifelong learners, adapt to new technologies, and approach ML with curiosity and a desire for continuous improvement.

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

Course Outcomes (COs)

COs	Description
M23BCSL606.1	Prepare data for machine learning by effectively applying preprocessing, feature engineering, and exploratory analysis techniques to diverse datasets.

M23BCSL606.2	Implement a range of machine learning algorithms by successfully building and training supervised and unsupervised learning models in Python, including k-NN, regression, decision trees, random forests, k-means, and PCA.
M23BCSL606.3	Analyze machine learning results by interpreting model outputs, identifying patterns and insights in data, and effectively communicating findings through visualizations and reports.
M23BCSL606.4	Evaluate and select appropriate machine learning models by critically comparing their performance using relevant metrics and considering factors beyond simple accuracy to justify choices for specific tasks.
M23BCSL606.5	Design a basic reinforcement learning agent by developing a Q-learning agent to solve a simple problem in a simulated environment.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCSL606.1	3	-	-	-	3	-	-	-	-	-	-	-	-	-
M23BCSL606.2	-	3	-	-	3	-	-	-	-	-	-	-	-	-
M23BCSL606.3	-	-	-	3	-	-	-	-	-	3	-	-	-	-
M23BCSL606.4	-	-	-	3	-	-	-	-	-	-	-	-	-	-
M23BCSL606.5	**-**	**-**	3	-	-	-	-	-	-	-	-	-	-	-
M23BCSL606	**3**	**3**	**3**	**3**	**3**	-	-	-	-	**3**	-	-	-	-

5. Assessment Plan (For PCL)

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**

10. Future with this Subject:

1. **Capstone Projects and Research:** The ML skills gained in this course can prove invaluable for final year projects. Students can leverage this knowledge to optimize robotic systems, analyze sensor data for predictive maintenance, or develop intelligent control algorithms — all empowered by the solid foundation built in this course.
2. **Advanced Engineering Electives:** Many higher-level electives, such as "Robotics," "Computer Vision," "Data Science," or "Artificial Intelligence," rely heavily on ML concepts. Students who have mastered the fundamentals through this course will be well-prepared to excel in these advanced subjects.
3. **Data-Driven Decision Making:** Regardless of their specialization, the ability to extract meaningful insights from data is an essential skill for any engineer. This course equips students to make informed, data-driven decisions, analyze trends, and approach complex engineering problems with a data-centric perspective.

Job Prospects:

1. **High Demand, High Reward:** ML expertise is highly sought after across industries, making it one of the most in-demand skillsets in today's job market. Graduates with a strong foundation in ML will have a competitive edge in securing rewarding and impactful roles.
2. **Career Versatility:** ML skills are transferable to a wide range of industries and job titles. Potential career paths include Machine Learning Engineer, Data Scientist, AI Specialist, Robotics Engineer, or any position requiring data analysis, predictive modeling, and intelligent system development.
3. **Future-Proof Skills:** The field of ML is constantly evolving. This course equips graduates with the foundational knowledge and adaptive learning mindset needed to stay ahead of the curve, embrace new advancements, and thrive in a dynamic career landscape.

6th Semester	**Ability Enhancement course(AEC)** **GENERATIVE AI LAB**	**M23BIS607A**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Understanding of Machine Learning**	Familiarity with fundamental machine learning concepts and techniques, including supervised and unsupervised learning, is essential. Understanding algorithms like linear regression, classification, and clustering provides a foundation for grasping more advanced generative models.
2	**Knowledge of Neural Networks**	A solid understanding of neural network architectures, including feedforward networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), is crucial. This background will help in comprehending how generative models like GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders) work.
3	**Programming Skills**	Proficiency in programming languages commonly used in AI development, such as Python, is necessary. Familiarity with libraries and frameworks like TensorFlow, PyTorch, or Keras will be beneficial for implementing and experimenting with generative models.
4	**Mathematical Foundations**	A strong grasp of mathematics, particularly linear algebra, calculus, and probability theory, is important. These mathematical concepts underpin many machine learning algorithms and generative models, enabling a deeper understanding of their workings and optimization.
5	**Experience with Data Handling and Preprocessing**	A strong grasp of mathematics, particularly linear algebra, calculus, and probability theory, is important. These mathematical concepts underpin many machine learning algorithms and generative models, enabling a deeper understanding of their workings and optimization.

2. Competencies

S/L	Competency	KSA Description
1	**Understanding Generative Models**	• Ability to select appropriate generative models based on specific use cases and data characteristics. • Ability to troubleshoot and optimize model performance to achieve desired outputs.
2	**Programming and Implementation**	• Ability to adapt and extend existing codebases for custom generative AI applications.
3	**Mathematical and Statistical Analysis**	• Ability to identify and address mathematical challenges during model development and training.
4	**Data Handling and Preprocessing**	• Ability to evaluate data quality and ensure it meets the requirements of generative models.
5	**Ethical and Practical Implications**	• Ability to make informed decisions regarding the ethical implications of generative AI projects.
6	**Research and Development**	• Ability to communicate the potential risks and benefits of generative AI technologies to stakeholders

3. Syllabus

Generative AI Lab (M23BIS607A) **SEMESTER – VI**			
Course Code	**M23BIS607A**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	0:0:2:0	SEE Marks	**50**
Total Number of Lecture Hours	**2 hr/week Practical**	Total Marks	**100**
Credits	**01**	Exam Hours	**01**

Course objectives: This course will enable students to:

- Understand and apply the fundamentals of Generative AI and large language models.
- Implement GPT architecture for text generation and dialogue systems.
- Fine-tune BERT for downstream NLP tasks and explore advanced Transformer architectures.
- Evaluate real-world applications, challenges, and emerging trends in Generative AI

List of problems for which student should develop program and execute in the Laboratory

1.	Generate a continuation of a given text prompt using GPT-2.
2.	Generate an image based on a text description using Stable Diffusion.
3.	Generate a short piece of music using Magenta.
4.	Apply style transfer to an image using a pre-trained neural network.
5.	Generate a summary of a long piece of text using a pre-trained summarization model.
6.	Create a simple chatbot that responds to user input using Rasa.
7.	Convert text into speech using Tacotron and a vocoder.
8.	Generate code snippets based on a prompt using OpenAI Codex.
9.	Generate an image of handwritten digits using a GAN.
10.	Generate a poem about a specific topic using GPT-3

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	Introduction, Program1
2	Week 3-4:	Program 2, Program 3.
3	Week 5-6:	Program 4, 5
4	Week 7-8:	Program 6, 7,8
5	Week 9-10:	Program 9,10

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 Testing live, showing step-by-step coding and debugging.
3	Lab Exercises	Design lab exercises that require students to implement and manipulate Testing process.
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.
7	Programming Assignments	Assign programming tasks and mini projects 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:

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**

1. SEE marks for practical course shall be 50 marks
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.

Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Build Generative Models	Learn to design and implement various generative models like GANs and VAEs.
2	Optimize Model Performance	Develop skills to evaluate and enhance the effectiveness of generative models.
3	Apply Generative AI	Apply generative AI techniques to practical problems in fields such as image and text generation.
4	Understand Ethical Implications	Analyze the ethical considerations and societal impacts of generative AI technologies.
5	Stay Updated with Trends	Keep up with the latest advancements and research in the field of generative AI.

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

Course Outcomes (COs)

COs	Description
M23BIS607A.1	Understand large language models' architecture and pre-training techniques
M23BIS607A.2	Apply the GPT and BERT model for natural language processing tasks
M23BIS607A.3	Analyzing the performance and limitations of large language models.
M23BIS607A.4	Examine the Generative AI practices.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BIS607A.1	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	2	-
M23BIS607A.2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	-	2
M23BIS607A.3	**-**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	-	3
M23BIS607A.4	**-**	**-**	**-**	**2**	**-**	**-**	**-**	**-**	**2**	**-**	**-**	**-**	-	-

M23BIS607A	**2**	**2**	**2**	**2**					**2**				2	2

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1-6	10	10	10	10	10	50
Total	10	10	10	10	10	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
Program 1-6	20	20	20	20	20	50
Total	20	20	20	20	20	**50**

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

1. **Design and Implement Generative Models**: Students will be able to design, develop, and implement various generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), using appropriate frameworks and tools.
2. **Conduct Research and Stay Current with Trends**: Students will be able to conduct research on current trends and advancements in generative AI, staying informed about the latest developments and innovations, and applying this knowledge to enhance their own projects and contributions to the field.
3. **Evaluate and Optimize Model Performance**: Students will be able to evaluate the performance of generative models using relevant metrics and techniques, and optimize models to enhance their effectiveness and efficiency in generating desired outputs.
4. **Apply Generative AI to Real-World Problems**: Students will be able to apply generative AI techniques to solve real-world problems across different domains, such as image synthesis, text generation, and data augmentation, demonstrating practical application skills.
5. **Analyze Ethical and Social Implications**: Students will be able to analyze and discuss the ethical and social implications of generative AI technologies, including issues related to bias, privacy, and misuse, and propose strategies for responsible AI development and deployment.

6th Semester	**Ability Enhancement-V(AE-V)** **Devops**	**M23BCS607B**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Understanding of Programming and Scripting Languages**	• Knowledge of programming fundamentals and experience with at least one scripting language (e.g., Python, JavaScript) is required to create automation scripts, work with Selenium WebdriverIO, and write scripts for Maven and Docker.
2	**Familiarity with Command Line Interfaces (CLI)**	• Proficiency in using command-line tools like GitBash or terminal commands is essential for interacting with Git, executing Docker commands, and managing CI/CD pipelines..
3	**Introduction to Software Development Life Cycle (SDLC)**	• A foundational understanding of the software development life cycle and its various stages will help students grasp the context of continuous integration, continuous deployment, and Agile methodologies used in DevOps practices.
4	**Basic Knowledge of Web Development and HTML/CSS**	• Familiarity with web technologies such as HTML and CSS is necessary for creating and understanding XPath expressions for locating elements in web applications during automation testing..
5	**Experience with Integrated Development Environments (IDEs) and Build Tools**	• Prior experience working with IDEs (e.g., IntelliJ IDEA, Eclipse) and build tools like Maven will be beneficial for managing project dependencies, building projects, and integrating with Jenkins for CI/CD..

2. Competencies

S/L	Competency	KSA Description
1	**Proficient Use of Version Control Systems**	**Knowledge:** Understand the principles and best practices of version control, including branching, merging, and conflict resolution.. **Skills:** Ability to create, manage, and collaborate on projects using Git, GitHub, and GitBash, including advanced operations like rebasing and resolving merge conflicts. **Attitudes:** Cultivate a disciplined approach to version control, emphasizing the importance of clear commit messages, regular updates, and collaborative workflows.
2	**Mastery of Continuous Integration and Automation Tools**	**Knowledge:** Understand the fundamentals of continuous integration, continuous deployment, and automation in a DevOps environment. **Skills:** Ability to configure and manage Jenkins pipelines, automate project execution, and apply various scheduling techniques within a CI/CD framework. **Attitudes:** Develop a proactive mindset towards automation, recognizing its role in improving efficiency, reducing errors, and ensuring consistent project delivery.
3	**Advanced Automation and Testing Techniques**	**Knowledge:** Comprehend the role of automated testing in software development, including the use of Selenium WebdriverIO and XPath. **Skills:** Ability to write and execute test scripts using Selenium WebdriverIO, effectively locate elements in web applications using different XPath strategies, and integrate with reporting tools for comprehensive test coverage. **Attitudes:** Foster an attention to detail and thoroughness in testing, emphasizing the importance of accuracy and reliability in automated testing processes.

4	**Integration of Build and Deployment Technologies**	**Knowledge:** Understand the lifecycle of software builds and deployments, including the use of Maven, Docker, and Jenkins. **Skills:** Ability to apply Maven lifecycle commands, createDocker containers, and integrate these tools with Jenkins to streamline the build and deployment processes. **Attitudes:** Embrace a continuous improvement mindset, focusing on optimizing build and deployment processes for speed, efficiency, and reliability.
5	**Effective Application of Agile and DevOps Practices**	**Knowledge:** Grasp the principles of Agile methodologies and DevOps practices, including backlog management, sprint planning, and the integration of various DevOps tools. **Skills:** Ability to create and manage project backlogs, sprints, and tasks using Agile techniques, retrieve and manipulate data using JQuery, and integrate multiple DevOps tools into a cohesive framework. **Attitudes:** Develop a collaborative and adaptable approach to project management, valuing teamwork, continuous feedback, and the iterative nature of Agile and DevOps practices.

3. Syllabus

DEVOPS
SEMESTER – VI

Course Code	**M23BCS607B**	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	**02**

Course Objectives:

1. Understanding and Application of Version Control Systems.
2. Mastering Continuous Integration and Automation.
3. Proficient Use of Automation Tools for Web Application Testing.
4. Integration and Management of Build and Deployment Pipelines
5. Implementation of Agile and DevOps Practices in Project Management.

Program -1

Demonstrate and Create project in local and remote repository using git and GitBash.

Program -2

Demonstrate and apply fork, merge, conflict and rebase concepts on repo using GitHub.

Program -3

Demonstrate the process of interfacing the repo using Jenkins to automate the project execution.

Program -4

Create a repository and apply different types of scheduling using continuous integration tool.

Program -5

Create a script to automate the operation using selenium WebdriverIO and integrate with reporting tools.

Program -6

Create a script using different types of Xpath and locate elements in an application.

Program-7

Apply maven life cycle commands on project and integrate with Jenkins tools

Program-8

Apply basic commands for Dockers and integrate with other tools.

Program-9

Create and apply backlogs, sprints, tasks for project and retrieve data using J-query.

Program-10

Create a Framework and integrate different DevOps tools for an application.

TEXTBOOKS:

1."Pro Git" by Scott Chacon and Ben StraubAvailable for free online.

2. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley

REFERENCE BOOKS:

1."Selenium WebDriver 3 Practical Guide" by UnmeshGundecha

2."Docker: Up & Running: Shipping Reliable Containers in Production" by Karl Matthias and Sean P. Kane

VIDEO LINKS:

1.https://www.youtube.com/watch?v=RGOj5yH7evk
2.https://www.youtube.com/watch?v=6YZvp2GwT0A

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1:** Introduction to Version Control and Git	Demonstrate and Create Project in Local and Remote Repository Using Git and GitBash Introduction to Git, setting up GitBash, creating local repositories, initializing remote repositories, and pushing code. Lab/Practice: Set up a Git repository, commit changes, and push them to GitHub.
2	**Week 2**: Advanced Git Concepts	Demonstrate and Apply Fork, Merge, Conflict, and Rebase Concepts on Repo Using GitHub Forking repositories, branching, merging, handling conflicts, and rebasing. Lab/Practice: Fork a repository, create a branch, make changes, merge, and resolve conflicts.
3	**Week 3:** Introduction to Jenkins and CI/CD	Demonstrate the Process of Interfacing the Repo Using Jenkins to Automate the Project Introduction to Jenkins, setting up a Jenkins server, and creating a basic CI/CD pipeline. Lab/Practice: Set up Jenkins, integrate it with a GitHub repository, and create an automated build.
4	**Week 4:** Advanced Jenkins and Scheduling	Create a Repository and Apply Different Types of Scheduling Using Continuous Integration Tools. Jenkins scheduling, cronjobs, and different scheduling strategies in CI/CD pipelines. Lab/Practice: Implement scheduling in Jenkins pipelines, using cron expressions for various tasks.
5	**Week 5:** Introduction to Selenium and Automation Scripting	Create a Script to Automate the Operation Using Selenium WebdriverIO and Integrate with Reporting Tools Introduction to Selenium WebdriverIO, writing basic test scripts, and integrating with reporting tools Lab/Practice: Write Selenium scripts using WebdriverIO, automate a basic web operation, and generate reports.
6	**Week 6:** XPath Strategies in Web Automation	Create a Script Using Different Types of XPath and Locate Elements in an Application Understanding XPath, types of XPath, and strategies for locating web elements. Lab/Practice: Create and run test scripts using different XPath strategies in a sample web application.
7	**Week 7:** Build Management with Maven	Apply Maven Life Cycle Commands on Project and Integrate with Jenkins Tools Introduction to Maven, Maven lifecycle phases, and integrating Maven with Jenkins. Lab/Practice: Create a Maven project, execute Maven lifecycle commands, and integrate with Jenkins for builds.
8	**Week 8:** Containerization with Docker	Apply Basic Commands for Docker and Integrate with Other Tools Introduction to Docker, Docker commands, containerization, and integration with CI/CD tools.

		Lab/Practice: Create and manage Docker containers, and integrate Docker with Jenkins in a CI/CD pipeline.
9	**Week 9:** Agile Methodologies and Project Management	Create and Apply Backlogs, Sprints, Tasks for Project and Retrieve Data Using JQuery. Agile methodologies, managing backlogs and sprints, using JIRA or similar tools, and using JQuery for data manipulation. Lab/Practice: Set up a project in an Agile tool, create and manage sprints and tasks, and retrieve data using JQuery.
10	**Week 10:** DevOps Framework and Tool Integration	Create a Framework and Integrate Different DevOps Tools for an Application Overview of DevOps frameworks, integrating tools like Jenkins, Docker, Maven, and Selenium into a cohesive CI/CD pipeline. Lab/Practice: Develop and demonstrate a full CI/CD pipeline integrating multiple DevOps tools in a project framework.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Project-Based Learning (PBL)	Integrate hands-on projects that align with each week's topics.Integrate hands-on projects that align with each week's topics.
2	Collaborative Learning and Peer Reviews	Encourage collaboration among students through pair programming, group tasks, and peer reviews. Organize students into small teams to work on projects, and incorporate peer review sessions where students evaluate each other's code, Git workflows, and automation scripts.
3	Flipped Classroom	Use the flipped classroom model, where students study theoretical content through videos and readings before class, and spend in-class time engaging in hands-on activities, discussions, and problem-solving.
4	Incremental and Iterative Learning	Teach concepts incrementally, building complexity over time. Start with foundational concepts like Git basics and gradually introduce more advanced topics like Jenkins integration, Docker commands, and full DevOpstoolchain integration.
5	Continuous Assessment and Feedback	Implement continuous assessment methods that include quizzes, coding challenges, and mini-projects.Conduct regular assessments such as short quizzes on Git commands, Jenkins pipelines, and Docker basics.

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**

FinalCIEMarks =(A) + (B)

SEE for practical Course:

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**

1. SEE marks for practical course shall be 50 marks
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	Master the Fundamentals of Version Control with Git	Students will be able to create, manage, and collaborate on projects using Git and GitHub, including setting up local and remote repositories, and executing advanced version control operations like branching, merging, rebasing, and resolving conflicts.
2	Implement Continuous Integration and Delivery Pipelines Using Jenkins	Students will learn to configure Jenkins for automating project builds, tests, and deployments, integrating it with GitHub repositories and other CI/CD tools, and applying various scheduling techniques to optimize workflow efficiency..
3	Analysis of Stress and StrainAutomate Web Application Testing Using Selenium WebdriverIO	Students will be able to write and execute automated test scripts using Selenium WebdriverIO, utilize different XPath strategies to locate web elements, and integrate these scripts with reporting tools for comprehensive test automation.
4	Apply Containerization and Build Management Techniques Using Docker and Maven	Students will gain proficiency in using Docker to containerize applications, apply Maven lifecycle commands for build management, and integrate these tools with Jenkins to streamline the build, test, and deployment processes..
5	Manage Agile Project Workflows and Retrieve Data Using JQuery	Students will learn to implement Agile methodologies, including creating and managing backlogs, sprints, and tasks in a project, and effectively use JQuery for data retrieval and manipulation within web applications..
6	Develop and Integrate a Comprehensive DevOpsToolchain Framework	Students will be able to design, develop, and implement a cohesive DevOpstoolchain framework that integrates multiple tools such as Git, Jenkins, Docker, and Maven, Selenium, and Agile project management practices to automate and manage the software development lifecycle efficiently.

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

Course Outcomes (COs)

COs	Description
M23BCS607B.1	**Apply Version Control Techniques** Demonstrate the ability to create, manage, and maintain local and remote repositories using Git and GitBash
M23BCS607B.2	**Implement Continuous Integration and Automation** Utilize Jenkins to automate the integration, testing, and deployment processes by creating and scheduling CI/CD pipelines.
M23BCS607B.3	**Develop Automation Scripts for Web Applications**
M23BCS607B.4	**Integrate Development Tools with Docker and Maven**
M23BCS607B.5	**Design Agile and DevOps Practices**.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS607B.1	3	3	-	-	3	-	-	-	-	-	-	-	3	-
M23BCS607B.2	3	3	3	-	3	-	-	-	-	-	-	-	-	3
M23BCS607B.3	3	-	3	-	3	3	-	-	-	-	-	-	3	-
M23BCS607B.4	3	3	3	3	3	-	-	-	-	-	-	-	-	3

M23BCS607B.5	3	3	3	3	3	3	-	-	3	-	-	-	3	3
M23BCS607B	**3**	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**3**	**-**	**-**	**-**	**3**	**3**

9. 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**

1.

2. 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**

10. Future with this Subject:

- **High Demand for DevOps Professionals:** Proficiency in tools like Git, Jenkins, Docker, and Selenium, coupled with Agile and DevOpsmethodologies, will make graduates highly sought after in industries ranging from tech startups to large enterprises.
- **Cloud-Native and Microservices Architectures**: Skills in containerization (e.g., Docker) and CI/CD pipelines will be critical for managing complex, distributed applications in cloud environments. Students trained in these areas will be well-prepared for roles in cloud engineering and site reliability engineering (SRE)
- **Automation and Artificial Intelligence in DevOps**: Knowledge in automation scripting and tool integration will be a foundational skill as AI-driven tools become standard in CI/CD pipelines and infrastructure management. This subject will prepare students to innovate in these areas.
- **Expansion of Agile and DevOpsBeyond IT**: A deep understanding of Agile and DevOps practices will be valuable not just in tech roles but also in broader organizational contexts, allowing graduates to contribute to the digital transformation of various industries.
- **Continuous Learning and Adaptability**: A deep understanding of Agile and DevOps practices will be valuable not just in tech roles but also in broader organizational contexts, allowing graduates to contribute to the digital transformation of various industries.

6th Semester	Programing with Kotlin	M23BIS607C

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Understanding of fundamental programming concepts such as 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 concepts like classes, objects, inheritance, polymorphism, and encapsulation
3	**Functional Programming Concepts**	Understanding functional programming concepts such as higher- order functions, immutability, pattern matching, and recursion
4	**Development Environment Setup**	Setting up a development environment for Kotlin programming is essential. IDEs such as IntelliJ IDEA, Eclipse, or Visual Studio Code with Kotlin plugins provide excellent support for development.

2. Competencies

S/L	Competency	KSA Description
1	**Conciseness**	**Knowledge:** Scala allows developers to write concise code, reducing boilerplate and enhancing productivity. **Skills:** Learning the basic syntax of Kotlin
2	**Functional Programming**	**Knowledge:** Scala provides strong support for functional programming constructs such as pattern matching, enabling developers to write clean and expressive code. **Skills:** Practicing the concept of functional programming
3	**Object- Oriented Programming**	**Knowledge:** Kotlin is fully 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.
4	**Concurrency:**	**Knowledge:** Kotlin provides powerful concurrency primitives such as actors and futures, making it well-suited for building scalable and concurrent applications. **Skills:** Learning relevant applications of Kotlin.
5	**Tooling and Ecosystem**	**Knowledge**: Kotlin 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.

3. Syllabus

Programing with Kotlin
SEMESTER – VI

Course Code	M23BIS607C	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**8-10 Lab slots**	Total Marks	**100**
Credits	**01**	Exam Hours	**2**

Course objectives: This course will enable students to:

- Understand basics of Kotlin programming.
- To gain an understanding of the processes that are involved in an Android Application Development
- Analyzing the complex problems with the Functional concept.
- Apply the new emerging tool to model the solution to various problems using Kotlin

PRACTICAL COMPONENT

. **1**	Installation of Kotlin.
2	Program for variable declaration in Kotlin
3	Program for declaring String and manipulating string.
4	Program on Operators

5	Program on Conditional Statement.
6	Program on Loops and Arrays
7	Program for Simple function and Recursive function.
8	Program on Default and Named argument in function.
9	Program for Inline function and Vararg parameter in function
10	Program for Classes and Objects
11	Program on Constructor.
12	Program for interfaces
13	Program for Exception handling
14	Program for App development.

Text Books:

1. "Kotlin in Action" Dmitry Jemerov, Svetlana Isakova -Manning Publications (2017)
2. "Kotlin for Android App Development" by Peter Sommerhoff.
3. Kotlin Notes For Professionals

References:

- JataTpoint
- Tutorialspoint

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-10: All lab Experiments	**Competency**: Kotlin programming **Knowledge:** Importance of Kotlin programming, ability to 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	Week 11-12: Exploring on IntelliJ Tool and Lab Internals, Assessment Evaluation	**Competency**: Understanding of tool usage **Knowledge:** Obtaining more information and knowledge on the tool usage. **Skills:** Analysing the problem to obtain the solution real time problems.

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	**Pair Programming**	Incorporate pair programming sessions where students collaborate in pairs to solve coding tasks or work on projects together.
7	**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) Continuous Internal Evaluation (CIE):
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:

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**

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

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Understanding Syntax and Basic Concepts:**	The fundamental syntax of Kotlin, including variables, data types, control structures, functions, and basic object-oriented concepts like classes and objects.
2	**Functional Libraries and Frameworks**	Exploring popular functional libraries and frameworks in the Kotlin ecosystem, such as Cats, Scalaz, and fs2, and understanding to use them to build functional, composable, and type-safe applications..
3	**Real-World Applications:**	Apply Kotlin knowledge to real-world projects or problem-solving exercises to gain practical experience and reinforce learning.
4	**Interoperability with Java**	Understanding the Kotlin interoperates with Java and learning best practices for integrating Kotlin code with existing Java libraries and frameworks.

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

COs	Description

M23BIS607C.1	Understand basics of Kotlin programming
M23BIS607C.2	Formulating the solution to complex problems with concepts of decision making, Looping and Inheritance
M23BIS607C.3	Designing the solution to the problems with the knowledge of List, tuple and File handling.
M23BIS607C.4	Analyzing the complex problems with the Functional concept.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO11	PO12	PSO 1	PSO 2
M23BIS607C.1	3													
M23BIS607C.2		3										3		
M23BIS607C.3			3									3	3	3
M23BIS607C.4				3									3	3
M23BIS607C	3	3	3	3	3							3	3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	CO5	**Total**
All Experiments	**10**	**10**	**10**	**10**	**10**	**50**
Total	**10**	**10**	**10**	**10**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**
All Experiments	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

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**: Functional programming paradigms are gaining popularity due to their suitability for building scalable, concurrent, and maintainable systems. Scala's strong support for functional programming makes it well-positioned to capitalize on this trend.
2. **Data Science and Big Data**: Scala is increasingly being used in the field of data science 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 in data science and big data is likely to expand further.
3. **Microservices Architecture**: Scala's support for building scalable and concurrent applications makes it well-suited for microservices architecture. As organizations increasingly adopt microservices to build modular, 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 features enable developers to express complex constraints and

enforce correctness at compile time, leading to more robust and scalable systems.

5. **Tooling and Ecosystem Growth**: The Scala ecosystem continues 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's interoperability with Java and its ability to leverage existing Java libraries and frameworks make it an attractive choice for organizations with Java-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.

6thSemester	**Ability Enhancement Course-V(AE-V)** **GOLANG PROGRAMMING**	**M23BCS607D**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Programming Knowledge	Understanding of programming concepts: Familiarity with basic concepts like variables, loops, conditionals, functions, and data structures (e.g., arrays, slices, maps) is helpful. Experience with another programming language: Prior experience with languages like Python, Java, or C can be beneficial, as it helps you understand Go's syntax and structure more easily.
2	Development Environment	Operating System: Go is cross-platform, so you can use it on Windows, macOS, or Linux. Text Editor or IDE: You can use any text editor like VS Code, Sublime Text, or a Go-specific IDE like GoLand. VS Code with the Go extension is highly recommended for Go development.
3	Go Compiler and Tools	**Git:** Familiarity with Git is helpful for version control, especially when working on larger projects or contributing to open source.
4	Command Line Interface (CLI)	Basic command-line knowledge: Understanding how to navigate your file system, execute commands, and manage packages using the CLI will be useful. Go often interacts with the terminal, especially when running or building programs.
5	Internet Connection	Access to documentation and packages: Go has extensive online documentation and a package ecosystem. You'll need an internet connection to access the Go documentation and to download third-party packages.
6	Familiarity with Go Conventions	Go coding standards: Go has specific coding conventions, like using tabs for indentation and organizing code in specific directory structures. Familiarizing yourself with these early on will help you write idiomatic Go code.

2. Competencies

S/L	Competency	KSA Description
1	Concurrency and Parallelism in Go	**Knowledge:** Understand Go's concurrency model, including goroutines, channels, and the Go scheduler. Familiarity with parallel processing and how Go manages concurrent tasks. **Skills:** Ability to design and implement concurrent applications that efficiently utilize multiple CPU cores. Proficiency in managing synchronization issues, avoiding race conditions, and ensuring thread safety. **Abilities:** Capable of analyzing and optimizing the performance of concurrent Go programs, ensuring that they are both efficient and scalable. Ability to debug complex concurrency-related issues.
2	Go Language Proficiency	**Knowledge:** Comprehensive understanding of Go's syntax, data types, control structures, and standard libraries. Familiarity with Go's memory management, garbage collection, and type system. **Skills:** Proficiency in writing clean, idiomatic Go code. Skilled in leveraging Go's built-in functions and packages to solve common programming tasks efficiently. **Abilities:** Able to apply Go's features to develop high-performance applications. Capable of mentoring others in Go's best practices and contributing to open-source Go projects.
3	Backend Development with Go	**Knowledge:** Understanding of backend architecture, RESTful API design, and database integration (both SQL and NoSQL) within the context of Go programming. Familiarity with web frameworks like Gin, Echo, or Fiber. **Skills:** Expertise in building and maintaining robust backend services and APIs using Go. Skilled in handling HTTP requests, routing, middleware, and integrating third-party services. **Abilities:** Capable of designing and implementing scalable backend systems, optimizing them for performance, and ensuring they can handle high loads. Ability to troubleshoot and resolve backend issues effectively.

4	Testing and Debugging in Go	**Knowledge:** In-depth knowledge of Go's testing framework, including testing package, benchmarking, and mock testing. Familiarity with Go's debugging tools and techniques. **Skills:** Proficient in writing unit tests, integration tests, and benchmarks to ensure code quality and performance. Skilled in debugging Go applications to identify and fix issues efficiently. **Abilities:** Able to design comprehensive test suites that cover all critical aspects of the application. Capable of using profiling tools to identify performance bottlenecks and optimize code accordingly.
5	Dependency Management and Project Structuring	**Knowledge:** Understanding of Go modules for dependency management, including versioning, package distribution, and workspace organization. Familiarity with Go's project structure and best practices. **Skills:** Ability to manage project dependencies effectively using Go modules. Skilled in organizing code into packages, managing external dependencies, and ensuring reproducible builds. **Abilities:** Capable of structuring large-scale Go projects in a way that promotes maintainability and scalability. Ability to resolve dependency conflicts and ensure smooth project collaboration.
6	Performance Optimization in Go	**Knowledge:** Deep understanding of Go's performance characteristics, including memory management, garbage collection, and CPU usage. Familiarity with tools for profiling and performance analysis. **Skills:** Expertise in identifying and optimizing performance bottlenecks in Go applications. Skilled in writing efficient code that minimizes memory usage and maximizes execution speed. **Abilities:** Able to conduct thorough performance audits and implement optimizations that improve the overall efficiency of Go applications. Ability to scale applications to meet increasing demands while maintaining performance.

3. Syllabus

GOLANG PROGRAMMING
SEMESTER – VI

Course Code	**M23BCS607D**	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	**02**

Course objectives: This course will enable students:

1. To learn the basics of Golang Programming .
2. To understand and develop Go applications using syntax, data structures, and standard libraries.
3. To Debug and analyze Go code for logic, performance, and concurrency issues.
4. To Design and optimize scalable backend services and APIs using Go best practices.

Pgm. No.	**List of Programs**
	PART-A
1	Design and Implement a Go program to print the name of the months and number of days based on user input number. Apply switch statement to implement the same.
2	Implement a calculator program that displays a menu with options 1. Add 2. Sub 3. Mul 4. Div Read 2 numbers and perform the relevant operation. After performing the operation, the program should ask the user if he wants to continue. If the user press Yes or Y, then the program should continue displaying the menu else the program should terminate.
3	Accept a n array of 5 positive integers. Create a program to find the smallest positive integer in the user input array which cannot be formed from the sum of 2 numbers in the array.
4	Develop a Go Program to check whether the user given matrix is a sparse or not.
5	Design and develop a simple Go function to find the longest substring without repeating characters in a given String.
6	Illustrate the different types of recursion in Go with suitable programs. Direct, Indirect, Tail and Head Recursion
	PART-B

7	Design a structure Employee with name and salary as its filed. Create three employee instances. Print the details and computer the average salary.
8	Create a program to swap two numbers using pointers in Go.
9	Apply pointer to structure concept to print the details of 3 student records. Assume Student record to contain USN, name and marks.
10	Develop a program to illustrate how to create an anonymous Goroutine.
11	Develop a program to illustrate how to start multiple Go routines.
12	Solve Producer Consumer concurrency issue using Go concurrency concept.

PART-C

Beyond Syllabus Virtual Lab Content (To be done during Lab but not to be included for CIE or SEE)

• Develop a Golang program to replace all occurrences of a word with another word in the given string. https://www.youtube.com/watch?v=vFqjpJfCG6Q

• Develop a calculator program using switch cases in Golang. https://www.youtube.com/watch?v=ca8xBxKWXsM

• Develop bubble sort implementation in Golang. https://www.youtube.com/watch?v=98yDJ5vao5Q

Suggested Learning resources

Text Books:

1. A Donovan, Brain W.Keringhan, “ The Go Programming Language”, Addison-Wesley Professional Computing Series,2016(Reprint)
2. An Introduction to Programming In Go by Caleb Doxsey

E-Reference Books:

1) www.tutorialgateway.org/go-programs
2) https://gobyexample.com

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-2:	**Program 1:** Design and Implement a Go program to print the name of the months and number of days based on user input number. Apply switch statement to implement the same. **Program 2:** Implement a calculator program that displays a menu with options 1. Add 2. Sub 3. Mul 4. Div Read 2 numbers and perform the relevant operation. After performing the operation, the program should ask the user if he wants to continue. If the user press Yes or Y, then the program should continue displaying the menu else the program should terminate.
2	Week 3-4:	**Program 3:** Accept a n array of 5 positive integers. Create a program to find the smallest positive integer in the user input array which cannot be formed from the sum of 2 numbers in the array. **Program 4:** Develop a Go Program to check whether the user given matrix is a sparse or not.
3	Week 5-6:	**Program 5:** Design and develop a simple Go function to find the longest substring without repeating characters in a given String. **Program 6:** Illustrate the different types of recursion in Go with suitable programs. Direct, Indirect, Tail and Head Recursion
4	Week 7-8:	**Program 7:** Design a structure Employee with name and salary as its filed. Create three employee instances. Print the details and computer the average salary. **Program 8**: Create a program to swap two numbers using pointers in Go.
5	Week 9-10:	**Program 9:** Apply pointer to structure concept to print the details of 3 student records. Assume Student record to contain USN, name and marks. **Program 10:** Develop a program to illustrate how to create an anonymous Goroutine.
6	Week 11-12:	**Program 11:** Develop a program to illustrate how to start multiple Go routines.

		Program 12: Solve Producer Consumer concurrency issue using Go concurrency concept.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	**Hands-on Coding Sessions**	Conduct regular coding exercises and labs where students actively write and run Go programs, reinforcing concepts learned in lectures. Provide practical projects that require the application of Go's concurrency model, standard libraries, and tools.
2	**Interactive Lectures and Discussions**	Use interactive lectures to introduce and explain Go's core concepts, such as goroutines, channels, and Go modules. Encourage class discussions and Q&A sessions to clarify doubts and deepen understanding of complex topics.
3	**Problem-Based Learning (PBL)**	Present real-world scenarios and problems that require students to design and develop solutions using Go.Assign group projects that foster collaboration and peer learning, simulating professional Go development environments.
4	**Code Reviews and Pair Programming**	Implement code review sessions where students evaluate each other's code, learning best practices and improving their coding style.Utilize pair programming to enhance collaboration, allowing students to work together to solve coding challenges.
5	**Quizzes and Assessments**	Use regular quizzes and assessments to gauge understanding of key Go concepts and reinforce learning.Include both theoretical questions and practical coding tasks in assessments to ensure comprehensive evaluation.
6	**Advanced Topics**	State Management: Introduce advanced topics like state management and how to handle complex data flows in larger applications.
7	**Continuous Feedback and Support**	Offer timely feedback on assignments and projects, guiding students in improving their Go programming skills.Provide additional support through office hours, online forums, or study groups to address individual learning needs.

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:

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**

1. SEE marks for practical course shall be 50 marks
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.

6. Learning Objectives

S/L	Learning Objectives	Description
1	Understand Go Syntax and Structure	Students will be able to identify and use Go's syntax, data types, and control structures to write basic programs
2	Implement Concurrency in Go	Students will understand Go's concurrency model and be able to create and manage goroutines and channels to develop concurrent applications.
3	Utilize Go's Standard Library	Students will learn to effectively use Go's standard library for tasks such as file handling, error management, and networking.
4	Develop and Test Go Applications	Students will be able to write, build, and run Go applications, and utilize Go's testing framework to create and run unit tests.
5	Design and Optimize Go-based Systems	Students will gain the ability to design, implement, and optimize scalable backend services and APIs using Go, focusing on performance and maintainability.
6	Apply Best Practices in Go Programming	Students will understand and apply Go programming best practices, including code organization, documentation, and code reviews.

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

Course Outcomes (COs)

Cos	Description
M23BCS607D.1	Apply the basic programming Go Lang constructs to develop standalone applications.
M23BCS607D.2	Apply the concept of functions and recursive functions in GoLang programming
M23BCS607D.3	Develop applications using Go Routines and channels
M23BCS607D.4	Solve the real-world concurrency issues using concurrency with go concepts.

`CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCS607D.1	**3**												**2**	
M23BCS607D.2		**3**											**2**	
M23BCS607D.3			**3**											**2**
M23BCS607D.4				**3**	**3**									**2**
M23BCS607D	**3**	**3**	**3**	**3**	**3**								**2**	**2**

8. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Program 1 to 12	**5**	**15**	**15**	**15**	50
Total	**5**	**15**	**15**	**15**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Program 1 to 12	**20**	**30**	**20**	**30**	100
Total	**20**	**30**	**20**	**30**	**100**

9. Future with this Subject

The future with Go (Golang) programming language looks promising, driven by its strengths and growing adoption in various industries. Here are some key trends and potential future developments:

1. **Increased Adoption in Cloud and Microservices:**
    - Go's efficiency and concurrency model make it ideal for cloud-native development and microservices. As more organizations move towards microservices architectures, Go's role in building scalable, high-performance services is likely to expand.
    - Companies like Google, Uber, and Dropbox have already integrated Go into their systems, and this trend is expected to grow, especially in cloud platforms like Kubernetes, which is itself written in Go.
2. **Growing Ecosystem and Tooling:**
    - The Go ecosystem is continuously evolving, with a growing number of libraries, frameworks, and tools. The introduction of modules for dependency management has made Go even more robust.
    - Tools like GoLand (IDE), Go's built-in testing suite, and profiling tools continue to improve, making Go development more accessible and efficient.
3. **Continued Focus on Simplicity and Performance:**
    - Go's design philosophy emphasizes simplicity and minimalism, which resonates with developers seeking to build maintainable and efficient systems. As the demand for performant and scalable software grows, Go's straightforward approach will remain attractive.
    - The language's performance characteristics, particularly in networked applications and large-scale systems, will keep it relevant in performance-critical environments.
4. **Expansion in DevOps and Site Reliability Engineering (SRE):**
    - Go's quick compilation times, static binaries, and cross-platform capabilities make it a preferred choice for developing DevOps tools and SRE applications. Its growing popularity in this area is likely to continue as infrastructure-as-code and automation become more prevalent.
5. **Broader Application in Emerging Technologies**:
    - As technologies like blockchain, AI, and IoT continue to grow, Go's reliability and performance could see it being adopted in these fields. For example, Go is already being used in blockchain platforms like Ethereum and Hyperledger Fabric.
    - The language's ease of deployment and low overhead make it suitable for resource-constrained environments like IoT devices.
6. **Enhanced Support for Web Development:**
    - Although not traditionally seen as a web development language, Go's use in backend web services and APIs is on the rise. Frameworks like Gin and Echo are helping Go gain traction in the web development community.
    - As web applications continue to evolve, Go could see increased use in performance-critical web backends and real-time services.
7. **Active Community and Language Evolution:**
    - The Go community is active and continues to drive the language forward with regular updates and proposals for new features. The community's focus on keeping Go simple yet powerful ensures that the language will continue to evolve to meet the needs of developers.
    - Future versions of Go are likely to introduce features that address current limitations while maintaining the language's core simplicity.
8. **Learning and Career Opportunities:**
    - As Go's popularity increases, there will be more opportunities for learning and career advancement in this field. Developers proficient in Go are already in demand, particularly in industries focused on cloud computing, backend development, and distributed systems.
    - Educational resources, both online and in academic settings, are likely to expand, further driving the language's adoption.

6th Semester	**Non-Credit Mandatory Course(NCMC)** **National Service Scheme(NSS)**	**M23BNSK608**

Non-Credit Mandatory Courses(NCMC) **National Service Scheme(NSS)**			
Course Code	**M23BNSK608**		
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**
Activities Report Evaluation by College NSS Officer at the end of every semester(3rd to 6thsemester)			

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. In addition 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 homework, 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 stakeholders–Implementation.
5. Preparing an action able business proposal for enhancing the village in come and approach for implementation.
6. Helping local schools to achieve good results and enhance their enrolment in 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 connect and 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:

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.

At the end of every semester, activity report should be submitted for evaluation.

Distribution of Activities – Semester wise from 3rd to 6th semester

Sem	Topics / Activities to be Covered
3rdSem for 25 Marks	• Organic farming, Indian Agriculture (Past, Present, and Future) Connectivity for marketing. • Waste management– Public, Private and Govt organization, 5R's. • Setting of the information imparting club for women leading to contribution in social and economic issues.
4thSem for 25 Marks	• Water conservation techniques– Role of different stakeholders–Implementation. • Preparing an actionable business proposal for enhancing the village income and approach for implementation. • Helping local schools to achieve good results and enhance their enrolment in Higher/technical/ vocational education.
5thSem for 25 Marks	• Developing Sustainable Water management systems for rural areas and implementation approaches. • Contribution to any national-level initiative of the Government of India. For eg. Digital India, Skill India, SwachhBharat, Atmanirbhar Bharath, MakeinIndia, Mudrascheme, Skill development programs etc. • Spreading public awareness under rural out reach programs.(minimum 5 programs). • Social connect and responsibilities.
6thSemfor 25 Marks	• Plantation and adoption of plants. Know your plants. • Organize National integration and social harmony events/workshops/seminars.(Minimum 02 programs). • Govt.school Rejuvenation and helping them to achieve good infrastructure.

Course outcomes (Course Skill Set):

COs	Description
M23BNSK608.1	Understand the importance of his/her responsibilities towards society.
M23BNSK608.2	Analyse the environmental and societal problems/issues and will be able to design solutions for the same.
M23BNSK608.3	Evaluate the existing system and to propose practical solutions for the same for sustainable development.
M23BNSK608.4	Implement government or self-driven projects effectively in the field.
M23BNSK608.5	Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

Pedagogy–Guidelines

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/ 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
2.	Waste management– Public, Private and Govt organization,5R's.	May be individual or team	Villages/ C ity Areas / Grama panchayat/ public	Site selection / proper consultation/ Continuous monitoring/ Information	Report should be submitted by an individual to the	Evaluation as per the rubrics of the scheme and

			associations/ Government Schemes officers /campus etc…	board	concerned evaluation authority	syllabus by NSS officer
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 / College 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
4.	Water conservation techniques – Role of different stakeholders– Implementation.	May be individual or team	Villages/ C ity 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
5.	Preparing an actionable business proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/ C ity 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
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 Schemesofficers /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
	Contribution to any			Group		

8.	national-level initiative of the Government of India. For eg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudrascheme, Skill development programs etc.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers /campus etc…	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 out reach programs.(minimum 5 programs).Social connect and responsibilities.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesofficers /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 Schemesofficers /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	Practice Session Description
1.	**Lecture session by NSS Officer**
2.	**Students Presentation on Topics**
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.	**Case-study-based Assessment, Individual performance**
11.	**Sector wise study and its consolidation**
12.	**Video-based seminar for 10-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.
- At last in 6th semester consolidated report of all activities from 3rd to 6thsemester, compiled report should be submitted as per the instructions.

Assessment Details:

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 the NSS Officer, the HOD, and the principal. • At-last Report should be 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.
Commencement 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 byeach student At the end 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 the field. There should be positive progress in the vertical order for the benefit of society in general.

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.

6th Semester	Non-Credit Mandatory Courses(NCMC) PHYSICAL EDUCATION (SPORTS & ATHLETICS) -III	M23BPEK608

Non-Credit Mandatory Course (NCMC)
PHYSICAL EDUCATION (SPORTS & ATHLETICS) -III

Course Code	**M23BPEK608**	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	**-**

Semester-V

PHYSICAL EDUCATION (SPORTS & ATHLETICS)—III

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

COs	**Description**
M23BPEK608.1	Understand the ethics and moral values in sports and athletics.
M23BPEK608.2	Perform in the selected sports or athletics of the student’s choice.
M23BPEK608.3	Understand the roles and responsibilities of organization and administration of sports and games.

Module-1

Ethics and Moral Values **(5hours)**

A. Ethics in Sports
B. Moral Values in Sports and Games

Module-2

Specific Games(Anyone to be selected by the student) **(20hours)**

A. Volley ball—Attack,Block,Service,Upper Hand Pass and Lower hand Pass.
B. Throwball—Service,Receive,Spinattack,NetDrop & Jumpthrow.
C. Kabaddi—Handtouch,ToeTouch,ThighHold,Anklehold and Bonus.
D. Kho-Kho—Giving Kho,SingleChain,Pole dive,Pole turning,3-6Up.
E. TableTennis—Service(ForeHand&BackHand),Receive(ForeHand&BackHand),Smash.
F. Athletics(Track/FieldEvents)—Any event as per availability of Ground.

Module-3

Role of Organisation and administration (5 hours)

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

6thSemester	Non-Credit Mandatory Course(NCMC) Yoga	M23BYOK609

Non-Credit Mandatory Courses(NCMC)
Yoga

Course Code	**M23BYOK609**		
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-Voce			

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
  4. Life with meaning, purpose, and direction
  5. Inner peace and tranquility
  6. Contentment

Yoga Syllabus

Semester V

- Patanjali's Ashtanga 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. Sheetali 3. Sheektari

Course outcomes (Course Skill Set):
At the end of the course, the student will be able to:

COs	**Description**
M23BYOK609.1	Understand the meaning, aim and objectives of Yoga.
M23BYOK609.2	Perform Suryanamaskar and able toTeach its benefits.
M23BYOK609.3	Understand and teach different Asanasbyname,its importance,methods and benefits.
M23BYOK609.4	Instruct Kapalabhati and its need and importance.
M23BYOK609.5	Teach different types of Pranayamaby its name,precautions,procedure and uses
M23BYOK609.6	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 (Twointernal 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