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

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Autonomous Institution Affiliated to VTU

Competency Based Syllabus (CBS)

for

Computer Engineering

(Under Outcome Based Education (OBE) and

Choice-Based Credit System (CBCS))

Offered from 5th to 6th Semesters of Study

in

Partial Fulfillment for the Award of Bachelor's Degree in

Computer Engineering

2023 Scheme

Scheme Effective from the academic year 2023-24

General Contents of Competency Based Syllabus Document

5th Semester	Professional Course (PC) THEORY OF COMPUTATIONS	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.
	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:** Understanding Formal Languages: Recognizing and working with different classes of formal languages, such as regular languages, context-free languages, and context-sensitive languages. **Skill:** Competency in Grammar Construction: Ability to construct and analyze grammars (regular, context-free) that Generates specific languages. **Attitude:** Language Properties: Using the properties of languages, such as closure properties, to reason about the languages and their classifications.
2	**Automata Theory**	**Knowledge:** Finite Automata: Designing deterministic (DFA) and nondeterministic (NFA) finite automata for recognizing regular languages. **Skill:** Pushdown Automata: Ability to design pushdown automata (PDA) to recognize context-free languages. Turing Machines: Designing Turing machines to model computation and recognize recursively enumerable languages.

		Attitude: Equivalence Proofs: proving the equivalence of different computational models (e.g., DFA and NFA, context-free grammars and PDA).
3	**Computability Theory**	**Knowledge:** Understanding Decidability**:** Determining whether a given problem is decidable or undecidable. **Skill:** Reductions and Mapping: Ability to reduce one problem to another to demonstrate undecidability or complexity classification. **Attitude:** Halting Problem: Understanding and proving the undecidability of the halting problem and its implications.
4	**Complexity Theory**	**Knowledge:** Complexity Classes: Understanding and differentiating between complexity classes such as P, NP, NP-complete, and NP-hard. **Skill:** Problem Classification: Ability to classify problems based on their computational complexity and understand the significance of reductions between problems. **Attitude:** P vs NP Problem: Understanding the importance of the P vs NP problem and its implications for computer science.
5	**Mathematical and Logical Reasoning**	**Knowledge:** Using formal proof techniques such as induction, contradiction, and diagonalization to prove results in the Theory of Computation. **Skill:**Ability to apply logical reasoning to formal systems, **Attitude:** ensuring correct formulation and analysis of computational problems.
6	**Applications and Implications**	**Knowledge:** Real-World Applications: Understanding the practical implications of theoretical results, such as the use of automata in lexical analysis or the impact of undecidability in software verification. **Skill:** programming languages, and artificial intelligence. **Attitude:** Interdisciplinary Connections: Competence in relating theoretical concepts to other areas of computer science, such as cryptography,

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)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course Objectives:

- Introduce core concepts in Automata and Theory of Computation
- Identify different Formal language Classes and their Relationships
- Design Grammars and Recognizers for different formal languages
- Prove or disprove theorems in automata theory using their properties
- 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**

RBT: L1, L2

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 **RBT: L1, L2,L3**
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** **RBT: L1, L2,L3**
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** **RBT: L1, L2,L3**
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** **RBT: L1, L2,L3**
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 andIntroduction 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, constructFinite State Transducers and Bidirectional Transducers.

3	Week 5-6: Regular Expressions Regular Grammars	Understand the concepts of Regular expressions 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. Understand the Properties of Context-Free 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.
6	Week 11:	Introduction to Turing machines, Variants of Turing machines (multitape, nondeterministic, etc.), understand the concept of Church-Turing thesisand Universal Turing machines.
7	Week 12:	Understand the concept of decidability and undecidability, Recursive and recursively enumerable languages, The Halting problem, the classes of P and NP, Time complexity and the class P,The class NP and NPcompleteness.

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) 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:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module.
- Marks scored will be proportionally scaled down to 50 marks **7. Learning Objectives**

S/L	Learning Objectives	Description
1	Understanding Formal Languages	Explain the concept of formal languages and distinguish between different classes of languages, such as regular, context-free, and contextsensitive 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	Acquire fundamental understanding of the core concepts in automata theory and Theory of Computation
M23BCS501.2	Design Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers
M23BCS501.3	Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness
M23BCS501.4	Classify a problem with respect to different models of Computation.

CO-PO-PSO Mapping

COs/POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
M23BIS501.1	-	3	-	-	-	-	-	-	-	-	-	2	3	-
M23BIS501.2	-	-	3	-	-	-	-	-	-	-	-	2	-	3
M23BIS501.3	-	-	3	-	-	-	-	-	-	-	-	2	-	3
23BIS501.4	-	-	-	2	-	-	-	-	-	-	-	2	-	2

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**2**	**8**			**10**
Module 2	**2**	**5**	**3**		**10**
Module 3	**2**	**8**			**10**
Module 4	**2**	**5**	**3**		**10**
Module 5	**4**			**6**	**10**
Total					**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	**Total**
Module 1	**5**	**15**			**20**
Module 2	**4**	**10**	**6**		**20**
Module 3	**5**	**15**			**20**
Module 4	**4**	**10**	**6**		**20**
Module 5	**4**	**8**		**8**	**20**
Total					**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.

6. **Blockchain and Cryptography**

- Protocol Verification: Automata theory is used to verify blockchain protocols, ensuring that they are secure and efficient. As blockchain technology grows, the demand for expertise in this area will increase.
- Cryptographic Algorithms: Automata theory aids in the design and analysis of cryptographic algorithms, which are essential for securing digital communications

5th Semester	Integrated Professional Course(IPC) COMPUTER NETWORK 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 itsdestination. **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:** Basicallyto 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: 1)**Identify sensitive or regulated data. 2) Understand how sensitive data is being accessed and shared. 3) 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:** IPSecurity 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 Network and Security **SEMESTER – V**			
Course Code	M23BCS502	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**	Total Marks	**100**
Credits	**04**	Exam Hours	**03**

Course Objectives:

- To understand network application architectures and the many types of service protocols such as HTTP, SMTP, FTP, and DNS.
- Analyze how the transport layer responds to network and application requests with various services and functions such as TCP and UDP.
- 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.
- Analyze security services and attacks by using data encryption and decryption techniques with the use of symmetric and asymmetric algorithms.
- 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 LinkState (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: Ch1.1 to 1.4, Ch2.1 to 2.3,Ch3.2,Ch5.1 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: Ch16.4 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.s

Part -B

1. 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.

2. Write a program to find the shortest path between vertices using bellman-ford algorithm.

3. Write a program for simple RSA algorithm to encrypt and decrypt the data

4. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.

TEXTBOOKS:

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 TEXTBOOK:

3. 1Computer-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 transportlayer 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)

Continuous Internal Evaluation

Components		Number	Weightage	Max. Marks	Min. Marks
Theory (A)	Internal Assessment-Tests (A)	2	**60%**	**15**	**06**
	Assignments/Quiz/Activity (B)	2	**40%**	**10**	**04**
	TotalMarks		**100%**	**25**	**10**

Components		Number	Weightage	Max. Marks	Min. Marks
Laboratory(B)	Record Writing	Continuous	**60%**	15	06
	Test at the end of the semester	1	**40%**	10	04
	TotalMarks		**100%**	**25**	**10**

FinalCIE Marks =(A) + (B) Semester

End Examination pattern:

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 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 applyapplication 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.

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 cybersecurity 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 cybersecurity 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.

5th Semester	**Professional Course (PC)** **INTERNET OF THINGS (IOT)**	**M23BCE503**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Electronics and Embedded Systems**	Understanding of Circuits: Knowledge of basic electronic components (resistors, capacitors, transistors, etc.). Microcontrollers: Familiarity with microcontrollers (e.g., Arduino, Raspberry Pi) and how to interface sensors and actuators with them. Embedded Programming: Experience with programming microcontrollers in languages like C or Python.

2	**Networking Basics**	Fundamentals of Networking: Understand concepts like IP addresses, DNS, TCP/IP, and how devices communicate over a network. Wireless Communication Protocols: Knowledge of wireless protocols such as Wi-Fi, Bluetooth, Zigbee, LoRa, and MQTT.
3	**Programming Skills:**	High-Level Languages: Proficiency in languages like Python, JavaScript, or C/C++. Scripting and Automation: Ability to write scripts for automation and data handling. Cloud Integration: Basics of cloud platforms (e.g., AWS, Azure) and how to connect IoT devices to the cloud.
4	**Data Handling and Processing**	Data Acquisition: How to collect data from sensors and IoT devices. Data Analysis: Basic understanding of data analysis, processing, and storage. Database Management: Experience with databases (SQL, NoSQL) to store IoT data.
5	**Operating Systems**	Linux: Basic knowledge of Linux/Unix-based systems, as many IoT devices run on Linux. Real-Time Operating Systems (RTOS): Familiarity with RTOS, especially for time-sensitive IoT applications.

2. Competencies

S/L	Competency	KSA Description
1	**Emergence of IoT**	**Knowledge:** understanding **IoT** evolution, the complex interdependence of enabling technologies, and the key networking components that support IoT systems. **Skills:** Analyzing how various technologies integrate and addressing strategies for efficient IoT deployment are essential. **Attitudes:**A forward-thinking attitude, curiosity about technological advancements, and a strategic approach to addressing challenges in IoT networking and addressing strategies will help in effectively navigating and contributing to the evolving IoT landscape.
2	**IoT Sensing and Actuation**	**Knowledge:** understanding various sensor types, their characteristics, and deviations, as well as different actuation methods and their properties. **Skills:** Designing and integrating sensors and actuators, handling sensor data, and calibrating devices for accurate readings applications. **Attitudes:** A thorough, detail-oriented approach and a proactive attitude towards troubleshooting and optimization for effectively implementing and managing sensing and actuation systems in IoT.
3	**IoT Processing Topologies and Types**	**Knowledge:** knowledge of data formats, real-time processing, and the trade-offs between edge and cloud processing. **Skills:** develop skills in programming, system design, and data processing, focusing on optimizing device selection and processing strategies. **Attitudes:** A curious, problem-solving mindset, attention to detail, and
		adaptability are essential attitudes for success in this dynamic field.

4	**IoT Processing Topologies and Types**	**Knowledge:** knowledge of various standards and protocols like IEEE 802.15.4, Zigbee, and LoRa, each with unique characteristics and use cases. **Skills:** Skills in selecting and implementing the appropriate technology based on factors like range, power consumption, and data rate are essential. **Attitudes:** A comprehensive, analytical mindset and adaptability are needed to evaluate and integrate these technologies effectively in diverse IoT environments.
5	**IoT Communication Technologies**	**Knowledge:** understanding various protocols like infrastructure, discovery, data, identification, and semantic protocols that facilitate device communication and management. **Skills:** Implementing and integrating these protocols, ensuring efficient data exchange and device management, are crucial for creating robust IoT systems. **Attitudes:** An analytical mindset, attention to detail, and a proactive approach to addressing interoperability challenges through standards and frameworks are essential for achieving seamless communication across diverse IoT devices and platforms.

3. Syllabus

INTERNET OF THINGS (IOT) SEMESTER – V			
Course Code	**23BCE503**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(4:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Courseobjectives: To teach students:

1. Thefundamentalsof Internet of Things and its building blocks along with their characteristics.
2. The recent application domains of IoT in everyday life.
3. The protocols and standards designed for IoT and the current research on it.
4. Associated technologies like cloud and fog computing in the domain of IoT.

Module -1

Emergence of IoT: Introduction, Evolution of IoT, Enabling IoT and the Complex Interdependence of Technologies, IoT Networking Components, Addressing Strategies in IoT.
Textbook 1: Chapter 4 – 4.1 to 4.5

Module -2

IoT Sensing and Actuation: Introduction, Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics.
Textbook 1: Chapter 5 – 5.1 to 5.9

Module -3

IoT Processing Topologies and Types: Data Format, Importance of Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading.
Textbook 1: Chapter 6 – 6.1 to 6.5

Module -4

IoT Connectivity Technologies: Introduction, IEEE 802.15.4, Zigbee, Thread, ISA100.11A, Wireless HART, RFID, NFC, DASH7, Z-Wave, Weightless, Sigfox, LoRa, NB-IoT, Wi-Fi, Bluetooth
Textbook 1: Chapter 7 – 7.1 to 7.16

Module -5
IoT Communication Technologies: Introduction, Infrastructure Protocols, Discovery Protocols, Data Protocols, Identification Protocols, Device Management, Semantic Protocols IoT Interoperability:
Introduction, Taxonomy of interoperability, Standards, Frameworks Textbook 1: Chapter 8 – 8.1, 6.2, 8.3, 8.4, 8.5, 8.6, .7 Textbook 1: Chapter 9 – 9.1, 9.2, 9.3
Textbook: 1. Sudip Misra, Anandarup Mukherjee, Arijit Roy, "Introduction to IoT", Cambridge University Press 2021. 2. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)",1st Edition, VPT, 2014. **Reference:** 1. S. Misra, C. Roy, and A. Mukherjee, 2020. Introduction to Industrial Internet of Things and Industry 4.0. CRC Press.
Weblinks and Video Lectures (e-Resources): 1. https://nptel.ac.in/noc/courses/noc19/SEM1/noc19-cs31/ Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: Emergence of IoT	Introduction, Evolution of IoT, Enabling IoT and the Complex Interdependence of Technologies, IoT Networking Components, Addressing Strategies in IoT.
2	Week 4-6: IoT Sensing and Actuation	Introduction, Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics.
3	Week 8-11: IoT Processing Topologies and Types	Data Format, Importance of Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading.
4	Week 7-8: IoT Connectivity Technologies	Introduction, IEEE 802.15.4, Zigbee, Thread, ISA100.11A, WirelessHART, RFID, NFC, DASH7, Z-Wave, Weightless, Sigfox, LoRa, NB-IoT, Wi-Fi, Bluetooth
5	Week 9-12: IoT Communication Technologies	Introduction, Infrastructure Protocols, Discovery Protocols, Data Protocols, Identification Protocols, Device Management, Semantic Protocols IoT Interoperability: Introduction, Taxonomy of interoperability, Standards, Frameworks

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 embedded system 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	Developing practical skills, understanding real-world constraints, and applying theoretical knowledge to tangible projects in embedded systems.

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:

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

7. Learning Objectives

S/L	Learning Objectives	Description
1	Emergence of IoT	Understanding the historical evolution and development of IoT, as well as the complex interdependence of enabling technologies and networking components. Additionally, students should be able to grasp addressing strategies and their role in the efficient implementation and scalability of IoT systems.
2	IoT sensing and actuation	Understanding the fundamentals of sensors and actuators, including their types, characteristics, and how they function in IoT systems. Additionally, students should be able to analyze sensor deviations, consider various sensing types, and make informed decisions about sensor and actuator selection based on application requirements.
3	IoT Processing Topologies and Types	Understanding the various data formats, the significance of data processing in IoT, and the different processing topologies such as edge and cloud computing. Students should also be able to evaluate IoT device design considerations and determine effective strategies for processing offloading to optimize system performance and resource management.
4	IoT connectivity technologies	Understanding the characteristics and applications of various communication protocols and standards such as IEEE 802.15.4, Zigbee, and LoRa, and how they impact IoT system design. Students should also be able to evaluate and select appropriate connectivity technologies based on factors like range, power consumption, and data requirements for different IoT applications.
5	IoT communication technologies and interoperability	Understanding the roles of various protocols (infrastructure, discovery, data, identification, and semantic) in enabling communication between IoT devices, as well as the importance of device management. Students should also grasp the taxonomy of interoperability and be able to apply relevant standards and frameworks to ensure seamless integration and communication across diverse IoT systems.

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

Course Outcomes (COs)

COs	Description
23BCE503.1	Understand the evolution of IoT, IoT networking components, and addressing strategies in IoT.

23BCE503.2	Analyze various sensing devices and actuator types.
23BCE503.3	Analyze the significance of processing in IoT.
23BCE503.4	Apply different connectivity technologies.
23BCE503.5	Understand the communication technologies, protocols and inter-operability in IoT.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
23BCE503.1	3	-	-	-	-	-	-	-	-	-	-	-	-	-
23BCE503.2	-	3	-	-	-	-	-	-	-	-	-	-	-	-
23BCE503.3	-	3	-	-	-	-	-	-	-	-	-	-	-	-
23BCE503.4	-	-	3	-	-	-	-	-	-	-	-	-	3	-
23BCE503.5	**3**	**-**	-	-	-	-	-	-	-	-	-	-	-	-
23BCE503	**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:

- IoT opens doors to a future where you can contribute to the development of smart, connected systems that transform industries like healthcare, manufacturing, and smart cities.
- With the growing demand for IoT solutions, there will be opportunities in areas such as IoT device development, network design, data analytics, and cybersecurity.
- As IoT continues to evolve, professionals with IoT expertise will play a critical role in shaping the integration of AI, edge computing, and 5G technologies.
- A career in IoT promises to be dynamic and impactful, driving innovation and efficiency in a connected world.

5th Semester	Professional Core Lab (PCL) IOT LAB	M23BCEL504

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Electronics	Understanding of basic electronic components (resistors, capacitors, diodes, transistors, etc.). Familiarity with circuits, voltage, current, and power concepts.
2	Programming Skills:	Proficiency in a programming language commonly used in IoT development, such as Python or C/C++. Knowledge of microcontroller programming (e.g., Arduino, ESP8266, ESP32).
3	Microcontroller/Microprocessor Fundamentals:	Understanding the architecture of microcontrollers (e.g., ARM, AVR, PIC). Knowledge of how to program microcontrollers and use development boards (e.g., Arduino, Raspberry Pi).
4	Networking Basics	Understanding of networking concepts like IP addressing, MAC addresses, and routing. Familiarity with communication protocols like HTTP, MQTT, and CoAP.
5	Sensor and Actuator Interfacing	Knowledge of how to interface sensors (temperature, humidity, motion, etc.) and actuators (motors, relays) with microcontrollers.
6	Operating System Knowledge	Basic understanding of Linux or other operating systems often used in IoT devices. Familiarity with real-time operating systems (RTOS) like FreeRTOS is a plus.

2. Competencies

S/L	Competency	KSA Description
1	**Interfacing I/O devices, sensors, and communication modules**	**Knowledge:** Understanding of different types of I/O devices and sensors such as LEDs, buzzers, push buttons, digital sensors (IR/LDR), and DHT11 sensors.Knowledge of communication modules like Bluetooth and their role in IoT. **Skill:** Ability to configure and use communication modules to send and receive data.Skill in reading and interpreting sensor data. **Attitude:**Methodical approach to integrating and testing sensor modules.Appreciation for the role of communication modules in IoT.Curiosity and willingness to learn about embedded systems and programming

2	**Remote monitoring of data and device control**	**Knowledge:** Understanding of cloud platforms like ThingSpeak and communication protocols like TCP.Knowledge of data uploading, retrieval, and real-time monitoring techniques. **Skill:**Ability to write aprogram to upload data to cloud platforms.Competence in retrieving and displaying data from the cloud.Skill in setting up servers (UDP/TCP) on Raspberry Pi and handling client requests. **Attitude:** Responsiveness to changes and updates in real-time data monitoring.
3	**Cloud Integration**	**Knowledge:** Understanding of IoT (Internet of Things) concepts and cloud services like ThingSpeak.Knowledge of how to retrieve and upload data to the cloud. **Skills:** Ability to write a program to interact with cloud platforms.Competence in using APIs to upload and retrieve data from ThingSpeak. **Attitudes:**Curiosity and willingness to learn about embedded systems and programming
4	**Networking and**	**Knowledge:** Understanding networking concepts and protocols (TCP, UDP, etc.)Understanding client-server architecture and communication models

	Communication Protocols	**Skill:** Ability to create and program TCP and UDP servers on the Arduino Uno. Ability to handle client requests and respond with appropriate data. **Attitude:**Willingness to learn and adapt to different communication protocols
5	**Interfacing I/O devices, sensors, and communication modules**	**Knowledge:** Understanding of different types of I/O devices and sensors such as LEDs, buzzers, push buttons, digital sensors (IR/LDR), and DHT11 sensors.Knowledge of communication modules like Bluetooth and their role in IoT. **Skill:** Proficiency in writing aprogram to interface with I/O devices and sensors.Ability to configure and use communication modules to send and receive data.Skill in reading and interpreting sensor data. **Attitude:**Methodical approach to integrating and testing sensor modules.Appreciation for the role of communication modules in IoT.Curiosity and willingness to learn about embedded systems and programming

3. Syllabus

IOT LAB SEMESTER – V			
Course Code	**23BCEL504**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(0:0:2:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**20 hours**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Courseobjectives:

The course aims to:

- Provide hands-on experience with Arduino Uno, a versatile single-board computer.
- Learn interfacing of various sensors and actuators with Arduino Uno.
- Develop skills in programming Arduino Uno for different applications.
- Introduce concepts of Internet of Things (IoT) and cloud integration.
- Enhance understanding of communication protocols (e.g., Bluetooth, UDP, TCP).

SL. NO	Experiments
1	Controlling the Light Emitting Diode (LED) /Buzzer with a push buttonusing Arduino Uno.
2	Controlling the LED blink rate with the potentiometer interfacing with Arduino
3	To interface OLED using Arduino uno andwritea C program to print temperature and humidity readings on it.
4	Directional Control of the DC motor using Arduino
5	Creation of Things Speak Account
6	To interface Bluetooth with ArduinoUno andwriteaC program to send sensor data to smartphone using Bluetooth.
7	Dht11sensor Data to Cloud
8	IotBased Air Pollution Control System
9	TdsSensor Interfacing with ArduinoUnousing Cprogram
10	Building Intrusion Detection System with Arduino and Ultrasonic Sensor
11	Mini project

Suggested Learning Resources:

1. Vijay Madisetti, Arshdeep Bahga, Internet of Things. "A Hands-on Approach", University Press
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, "Introduction to Internet of Things: A practical Approach", ETI Labs

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-4: Experiments 1 to 4	Interface LED, Buzzer, Pushbutton, LDR, and IR sensors, DHT11 sensor, OLED, and Relay with motor with ArduinoUno using C programfor different applications.
2	Week 5-6:	Interface Bluetooth of the smartphone with ArduinoUno for monitoring and
	Experiments 5 ,6 and 7	control of temperature sensed by DHT11 sensor. Uploading and retrieving sensor data to/from the cloud using IOT platform Thing Speak
3	Week 7-8: Experiments 8 & 9	Iot Based Air Pollution Control System And Tds Sensor Interfacing with Arduino Unousing C program
4	Week 9-10: Experiments 10 & mini project	Building Intrusion Detection System with Arduino and Ultrasonic Sensor. Building the IOT based real time system.

5. Teaching-Learning Process Strategies

TLP Strategies:	Description
Lecture Method	Utilize various teaching methods within the lecture format to reinforce competencies.
Video/Animation	Incorporate visual aids like videos/animations to enhance understanding of concepts.
Collaborative Learning	Encourage collaborative learning for improved competency application.
Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
Laboratory Learning	Developing practical skills, understanding real-world constraints, and applying theoretical knowledge to tangible projects in embedded systems.

6. Assessment Details (both CIE and SEE)

Continuous Internal Evaluation: Marks distribution for Program based Practical Course for CIE

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

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

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

1. SEE marks for practical course shall be 50 marks

Marks distribution for Experiment based Practical Course for Final CIE

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

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

7. Learning Objectives

S/L	Learning Objectives	Description
1	Interfacing Input/Output Devices	Students will understand the principles of interfacing LEDs, buzzers, push buttons, relay, motor, and digital sensors (IR, LDR) with the Arduino Uno.
2	Sensor Data Acquisition and	Students will learn to interface and read data from analog sensors like temperature
	Display	and humidity sensors (DHT11) using Arduino Uno. Students will develop programs to display sensor data on OLED displays or other output devices connected to the Arduino Uno.
3	Proficiency in using C programSyntax	Students will become proficient in writing a code on ArduinoUno for various applications
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	Interfacing Input/Output Devices	Students will understand the principles of interfacing LEDs, buzzers, push buttons, relay, motor, and digital sensors (IR, LDR) with the Arduino Uno.

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

Course Outcomes (COs)

COs	Description
M23BCEL504.1	Explain the fundamental concepts and working principles of sensors, actuators, and communication modules used with ArduinoUno.
M23BCEL504.2	Develop and implement using C programto interface various sensors and actuators with ArduinoUno for specific tasks like monitoring and control.
M23BCEL504.3	Create network-based applications using different protocols for data exchange and remote monitoring of embedded systems

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
M23BCEL504.1	3											
M23BCEL504.2	3	3	3		2				2	2		2
M23BCEL504.3	3	3	3		2				2	2		2
M23BCEL504	**3**	**3**	**3**		**2**				**2**	**2**		**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

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

Total	**5**	**10**	**10**	**5**	**20**	**50**

Semester End Examination (SEE)

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

Conditions for SEE Paper Setting:

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

10. Future with this Subject:

- The future of studying IoT labs using Arduino Uno will focus on creating affordable and accessible handson experiences for students.
- Arduino Uno, with its user-friendly platform, will continue to be a popular choice for prototyping IoT projects, enabling students to build and test real-world applications.
- As IoT evolves, there will be a push to integrate more advanced sensors and wireless communication modules, such as Bluetooth and Wi-Fi, with Arduino-based systems. Educational resources and tutorials will increasingly emphasize IoT security and data handling, preparing students for the challenges of the connected world.
- Open-source communities will play a key role in expanding the capabilities of Arduino Uno for IoT labs, fostering innovation and collaboration.
- Smart Home and Home Automation: IOT with Arduino Uno can be used to develop intelligent home automation systems, allowing users to control and monitor various aspects of their homes, such as lighting, temperature, security, and appliances, through a centralized system or mobile applications.
- Industrial Automation and Control: The combination of Arduino Uno and IOT can be leveraged in industrial settings for automation, process control, and monitoring applications. These systems can be used for tasks such as data acquisition, machine control, and predictive maintenance, leading to increased efficiency and productivity.

5th Semester	Professional Elective (PE) COMPUTER GRAPHICS	M23BCE505A

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

SEMESTER – V COMPUTER GRAPHICS			
Course Code	**M23BCE505A**	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

Course Objectives:

- To, learn hardware, software and OpenGL Graphics Primitives
- To analyze interactive computer graphics using OpenGL
- To design and implement algorithms for 2D graphics Primitives and attributes.
- To Demonstrate Geometric transformations, viewing on both 2D and 3D objects
- 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, Video Display Devices: 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, curve attributes, OpenGL point attribute functions, OpenGL line attribute functions, Line drawing algorithms(DDA, Bresenham"s), circle generation algorithms (Bresenham"s).

Module -2

2D Geometric Transformations and 2D viewing:

2DGeometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2DComposite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, and OpenGL geometric transformations function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.

Module -3

Clipping, 3D Geometric Transformations, Color and Illumination Models:

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.3DGeometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, **OpenGL** geometric transformations functions.

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..

Module -5

Input & interaction, Curves and Computer Animation:

Input and Interaction: Input devices, clients and servers, Display Lists, Display Lists and Modeling, Programming Event Driven Input, Menus Picking, Building Interactive Models, Animating Interactive programs, Design of Interactive programs, Logic operations .Curved surfaces, quadric surfaces, OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding openGL functions.

TEXTBOOKS:

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

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

REFERENCE BOOKS:

James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: pearson education

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, Video Display Devices: Random Scan and Raster Scan displays, graphics software. OpenGL: Introduction to OpenGL , OpenGL point functions, OpenGL line functions, OpenGL line attribute functions, Line drawing algorithms (DDA, Bresenham"s), circle generation algorithms (Bresenham"s).
2	Week 4-5: **Fill area Primitives ,2D Geometric transformations and 2D viewing**	Understanding 2DGeometric Transformations: Basic 2D Geometric Transformations, 2DComposite transformations, other 2D transformations raster methods for geometric transformations, OpenGL raster transformation.
3	Week 6-8: **Clipping, 3D Geometric Transformations, Colour and Illumination models**	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. 3DGeometric Transformations **OpenGL** geometric transformations functions. .
4	Week 9-10: **3D viewing and Visible Surface Detection**	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, Curves and Computer Animation**	Input and Interaction: Input devices, clients and servers, Display Lists, Building Interactive Models, Animating Interactive programs \OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions.

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	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.

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:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module.
- Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	**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

Course Outcomes (COs)

- Design and implement algorithms for 2D graphics primitives and attributes.
- Illustrate Geometric transformations on both 2D and 3D objects.
- Apply concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
- Decide suitable hardware and software for developing graphics packages using OpenGL.

COs	Description
M23BCE505A.1	**Understand and Apply** fundamental concepts of computer graphics 2D graphics primitives and attributes
M23BCE505A.2	**Analyze** different Geometric transformations on both 2D and 3D objects.
M23BCE505A.3	**Design** and implement algorithms of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
M23BCE505A.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
M23BCE505A.1	3		-	-	-	-	-	-	-	-	-	-	3	-
M23BCE505A.2	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCE505A.3	-	-	3		-	-	-	-	-	-	-	-	3	-
M23BCE505A.4	-	-	-	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**	**10**	**50**

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	Total
Module 1	**20**				**20**
Module 2		**20**			**20**
Module 3			**20**		**20**
Module 4				**20**	**20**
Module 5				**20**	**20**
Total	**20**	**20**	**20**	**20**	**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 highquality 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 (PE) CLOUD COMPUTING	M23BCE505B

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Familiarity with at least one programming language (e.g., Python, Java, or JavaScript).). Understanding of basic programming concepts like variables, loops, conditionals, functions, and data structures.
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:** to problem-solving and project management capabilitie. **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 cloudbased 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 Security**	**Knowledge:** Understanding the Mastering cloud security requires a deep
		understanding of a wide range of concepts, from identity and access management to data protection, network security, and compliance.. **Skills:** Ablility to Cloud Platform Expertise**,** Identity and Access Management, Network Security, Encryption and Data Protection **Attitudes:** cloud security is crucial as they shape how security challenges are approached.
5	**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.

3. Syllabus

CLOUD COMPUTING SEMESTER – V			
Course Code	**M23BCE505B**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(3:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Course Objectives:
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.

Module -1

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
Textbook 1: Chapter 1: 1.1,1.2 and 1.3

Module -2

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, Technology Examples.
Textbook 1 : Chapter 3: 3.1 to 3.6

Module -3

Cloud Computing Architecture: Introduction, Cloud Reference Model, Types of Clouds, Economics of the Cloud, Open Challenges. **Textbook 1: Chapter 4: 4.1 to 4.5**
Module -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, Observations. **Textbook 1: Chapter 9: 9.1 to 9.2**
Module -5
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. **Textbook 1: Chapter 10: 10.1 to 10.2**

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. Toby Velte, Anthony Velte, Cloud Computing: A Practical Approach, McGraw-Hill Osborne Media.
2. George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Publication.

Weblinks and Video Lectures (e-Resources):

1. https://www.youtube.com/watch?v=1N3oqYhzHv4
2. 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 7-8:	**Cloud Computing Architecture:** Introduction, Cloud Reference Model, Types of Clouds, Economics of the Cloud, Open Challenges
4	Week 9-10:	**Cloud Security**: Risks, Top concern for cloud users, privacy impact assessment, trust, OS security, VM Security, Security Risks posed by shared images and management OS
5	Week 11-12:	**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, Observations.

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 of 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:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module. • Marks scored will be proportionally scaled down to 50 marks

13. 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 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 Security	Risks, Top concern for cloud users, privacy impact assessment, trust, OS security, VM Security, Security Risks posed by shared images and management OS
5	Cloud Platforms in Industry	Compute services, Storage services, Communication services, Additional services. Google AppEngine: - Architecture and core concepts, Application life cycle, Cost model, Observations

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

Course Outcomes (COs)

COs	Description
M23BCE505B.1	Compare cloud computing environment utilized for real time applications.
M23BCE505B.2	Identify various models of cloud computing.
M23BCE505B.3	Analyze how to design cloud native applications.
M23BCE505B.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
M23BCE505B.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE505B.2	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE505B.3	3	3	-	3	-	3	-	-	-	-	-	3	3	3
M23BCE505B.4	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE505B	3	-	-	-	-	-	-	-	-	-	-	-	3	3

15. 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**

16. 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 slicing.

5th Semester	Professional Elective (PE) EMBEDDED SYSTEMS	M23BCE505C

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Mathematics:**	Understanding of basic mathematical concepts, including Boolean algebra and binary arithmetic. Knowledge of control systems and signal processing is a plus for more advanced embedded systems work.
2	**Basic Electronics and Circuit Theory:**	Understanding of analogy and digital circuits. Familiarity with components like resistors, capacitors, transistors, and diodes. Knowledge of circuit design and analysis.
3	**Programming Skills:**	Proficiency in C/C++ is essential, as these languages are commonly used in embedded systems programming. Familiarity with assembly language can be beneficial, especially for low-level programming. Basic knowledge of Python or other scripting languages for automation and testing can be useful.
4	**Microcontroller/Microprocessor Fundamentals:**	Understanding the architecture of microcontrollers (e.g., ARM, AVR, PIC). Knowledge of how to program microcontrollers and use development boards (e.g., Arduino, Raspberry Pi).
5	**Computer Architecture:**	Basics of how processors work, including concepts like registers, memory hierarchy, and instruction sets. Understanding of real-time operating systems (RTOS) is helpful.

6	**Digital Logic Design:**	Knowledge of combinational and sequential logic circuits. Understanding of how to design and implement finite state machines (FSMs).
7	**Communication Protocols:**	Familiarity with serial communication protocols like UART, SPI, I2C. Knowledge of networking protocols like TCP/IP if dealing with networked embedded systems.
8	**Operating Systems:**	Basic understanding of operating system concepts, particularly those related to embedded systems (e.g., task scheduling, interrupt handling).

2. Competencies

S/L	Competency	KSA Description
1	**Core of Embedded Systems**	**Knowledge:** Understanding the internal workings, including the CPU, memory, and peripheral interfaces. Knowledge of different types of microcontrollers (e.g., ARM Cortex, AVR, PIC) and their applications. **Skills:** Ability to integrate hardware and software components, ensuring they work together seamlessly. And Skill in developing and testing prototypes, and iterating designs based on feedback and test results. **Attitudes:** A meticulous approach to designing, coding, and testing embedded systems, where small errors can lead to significant problems.
2	**Characteristics and quality attitude of embedded systems**	**Knowledge:** knowledge of critical factors like real-time performance, reliability, and resource constraints, essential for designing effective systems. **Skills:** Abilityin optimizing these attitudes through careful design, testing, and validation, ensuring that systems meet their specific requirements. This
		field also hones your ability to perform trade-offs between competing quality attitude, such as balancing performance with power efficiency. **Attitudes:** Ameticulous attention to detail and a strong commitment to quality, ensuring that every aspect of the system is robust and reliable. Cultivate critical thinking and an ethical mindset, necessary for designing systems that are safe, secure, and dependable in real-world applications.
3	**Hardware Software Co design and Program Modelling**	**Knowledge:** knowledge of how to integrate hardware and software components seamlessly, optimizing system performance and efficiency. **Skills:** Ability to designing and modelling complex systems, using tools like HDL (Hardware Description Languages) for hardware and high-level languages for software, ensuring they work together harmoniously. This field also enhances your ability to perform trade-off analysis, balancing computational tasks between hardware and software for optimal results. **Attitudes:** Strong analytical thinking, attention to detail, and adaptability, enabling you to tackle the dynamic challenges of co-design.
4	**Embedded Firmware Design**	**Knowledge:** knowledge of low-level programming, memory management, and hardware interfacing, crucial for creating reliable firmware for embedded systems. **Skills:** Ability in writing efficient, hardware-specific code in languages like C/C++, and using debugging tools to ensure the firmware operates as intended. This field also enhances your ability to optimize firmware for performance and power efficiency while maintaining stability. **Attitudes:**Precision and patience, essential for dealing with the intricate details of firmware development

5	**Real-time Operating System (RTOS)based Embedded System Design:**	**Knowledge:** Real-time Operating System (RTOS) based Embedded System Design provides you with knowledge of how RTOS manages tasks, prioritizes processes, and handles real-time constraints in embedded systems. **Skills:** Develop skills in configuring and programming RTOS to ensure that critical tasks are executed within specified time limits, essential for applications like automotive systems and industrial automation. **Attitudes: C**ommitment to precision and reliability, as real-time systems often operate in critical environments where timing and stability are paramount.

3. Syllabus

EMBEDDED SYSTEMS SEMESTER – V			
Course Code	**M23BCE505C**	CIE Marks	**50**
Number of Lecture Hours/Week (L: T: P: S)	**(3:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**40 hours Theory**	Total Marks	**100**
Credits	**03**	Exam Hours	**03**

Courseobjectives: To teach students

- Introductory topics of Embedded System design
- Characteristics & attitude of Embedded System
- Introduction of Embedded System Software and Hardware development
- RTOS based Embedded systemdesign

Module -1

Introduction to Embedded system: Embedded systems versus General computing systems, classification of Embedded systems, applications of embedded systems, purpose of embedded systems, core of embedded system, memory, sensors and actuators, Communication interface, other system components.
Textbook 1: 1.2, 1.4, 1.5, 1.6, 2.1, 2.2.1, 2.2.2, 2.3, 2.4

Module -2

Attributes of Embedded system: Characteristics of embedded system, quality attributes of embedded system, washing machine- application specific embedded system, automotive- domain specific examples of embedded system, factors to be considered in Selecting a Controller, Embedded system development environment.
Textbook 1: 3.1, 3.2, 4.1, 4.2, 5.1, 13.1

Module -3

Hardware Software Co design and Program Modelling: Fundamental issues in Hardware SoftwareCodesign,ComputationalmodelsinEmbeddedSystemDesign .
Chapter7–Text1:7.1,7.2
EmbeddedHardwareDesignandDevelopment: Analog Electronic Components,DigitalElectronic Components, VLSI & Integrated Circuit Design, Electronic Design Automation Tools
Textbook1:8.1,8.2,8.3, 8.4

Module -4

Embedded Firmware design: Embedded Firmware Design Approaches, Embedded Firmware Development Languages, programming in Embedded C. Textbook 1: 9.1, 9.2, 9.3

Module -5

RTOS based Embedded system design: Types of operating system, tasks, process and threads, multiprocessing and multitasking, task communication, task synchronization, device drivers, how to choose an RTOS. Textbook 1: 10.2, 10.3, 10.4, 10.5, 10.7, 10.8, 10.9, 10.10

TEXTBOOK: **1.** ShibuKV,"IntroductiontoEmbeddedSystems",SecondEdition,McGrawHillEducation **2.** Embedded systems by Rajkamal, McGraw Hill, 2nd Edition
Weblinksand VideoLectures(e-Resources):
NPTLLectures:https://nptel.ac.in/courses/108102045Embedded Systems, IIT Delhi, Prof. Santanu Chaudhary

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: Introduction & Core of Embedded Systems	Introduction to Embedded Systems and general-purpose computer systems, history, classifications, applications and purpose of embedded systems.Microprocessors and microcontrollers, RISC and CISC controllers, Big endian and Little-endian processors, Application specific ICs, Programmable logic devices, COTS, sensorsandactuators,communicationinterface,embeddedfirmware,other systemcomponents,PCBandpassivecomponents
2	Week 4-6: Characteristicsandqualityattributesofembeddedsystems	The Characteristics,Operationalandnonoperational quality attributes, application specific embedded system - washing machine, domain specific – automotive
3	Week 8-11: Hardware Software Co design and Program Modelling	Fundamental issues in Hardware SoftwareCo-design,ComputationalmodelsinEmbeddedSystemDesign Analog Electronic Components,DigitalElectronic Components, VLSI & Integrated Circuit Design, Electronic Design Automation Tools
4	Week 7-8: EmbeddedFirmwareDesign	EmbeddedFirmwareDesignApproaches, Embedded Firmware Development Languages, Types of files generated on cross compilation (only explanation–programmingcodesneednotbedealt),disassemble/decompliler,Simulators,EmulatorsandDebugging
5	Week 9-12: RTOS based EmbeddedSystemDesign	Operating System basics, Types of Operating Systems, Tasks, Process and Threads, Multiprocessing and Multitasking, Task Scheduling.

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 embedded system 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	Developing practical skills, understanding real-world constraints, and applying theoretical knowledge to tangible projects in embedded systems.

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:

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

7. Learning Objectives

S/L	Learning Objectives	Description
1	**Introduction & Core of Embedded Systems**	Understand the fundamental concepts of embedded systems, including their architecture, components, and typical applications. Core learning objectives focus on gaining practical skills in designing and implementing embedded systems, including programming microcontrollers, interfacing with hardware, and optimizing system performance for real-world applications.
2	**Characteristicsandqualityattributesofembeddedsystems:**	Understand the key factors that define system performance, reliability, and efficiency, and how these attributes impact design decisions. Additionally, students aim to develop the skills to evaluate and optimize embedded systems to meet specific requirements and constraints, ensuring they deliver high-quality, dependable performance in various applications.
3	**Hardware Software Co design and Program Modelling & EmbeddedHardwareDesignandDevelopment**	Understand the principles of integrating hardware and software components effectively and to develop skills in designing and modelling complex systems to optimize performance and functionality. For Embedded Hardware Design and Development, the objectives are to gain expertise in designing, prototyping, and
		testing hardware components and systems, ensuring they work seamlessly with embedded software to meet specified requirements and constraints.
4	**EmbeddedFirmwareDesignandDevelopment**	Master the creation of low-level, hardware-specific code, including writing, debugging, and optimizing firmware for embedded systems. For Embedded System Development Environments, the objectives are to gain proficiency in using various development tools, environments, and methodologies to streamline the design, implementation, and testing of embedded systems.

5	**Real-timeOperatingSystem (RTOS)basedEmbeddedSystemDesign**	Understand the principles of real-time scheduling, task management, and system constraints to develop reliable and responsive embedded systems. Additionally, students aim to gain skills in configuring and optimizing RTOS for efficient resource management and meeting strict timing requirements in real-time applications.

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

Course Outcomes (COs)

COs	Description
M23BCE505C.1	Understand the characteristicsofEmbeddedSystem design
M23BCE505C.2	Acquireknowledgeaboutbasic conceptsofcircuitemulators,debuggingandRTOS
M23BCE505C.3	Analyseembeddedsystemsoftwareandhardwarerequirements
M23BCE505C.4	Developprogrammingskillsinembeddedsystemsforvarious applications.
M23BCE505C.5	Designbasic embeddedsystemforrealtime applications

CO-PO-PSO Mapping

COs/Pos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCE505C.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCE505C.2	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCE505C.3	-	-	3	3	-	-	-	-	-	-	-	-	3	-
M23BCE505C.4	-	-	-	3	-	-	-	-	-	-	-	-	-	3
M23BCE505C.5	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCE505C	**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:

- **Artificial Intelligence and Machine Learning:** Integration of AI and machine learning into embedded systems will enable more intelligent and adaptive devices, capable of real-time decision-making and predictive maintenance.

- **Edge Computing:** Embedded systems will play a crucial role in edge computing by processing data locally on devices, reducing latency and bandwidth usage while improving performance and security.
- **Advanced Connectivity:** With the rollout of 5G and future communication technologies, embedded systems will benefit from enhanced connectivity, supporting faster and more reliable data exchange between devices.
- **Autonomous Systems:** The development of autonomous vehicles, drones, and robotics will rely heavily on embedded systems for real-time control, sensor integration, and complex decision-making processes.
- **Wearable Technology:** Continued innovation in wearable technology will push the boundaries of embedded systems, focusing on health monitoring, fitness tracking, and augmented reality.
- **Security and Privacy:** As embedded systems become more ubiquitous, there will be a growing emphasis on cybersecurity to protect against vulnerabilities and ensure data privacy in connected devices.
- **Sustainability:** Embedded systems will contribute to sustainability efforts through energy-efficient designs, smart grids, and environmental monitoring, helping to address global challenges such as climate change.

5th Semester	**Professional Elective (PE)** **ADVANCED JAVA**	**M23BCE505D**

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	**M23BCE505D**	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	**04**	Exam Hours	**03**

Course Learning Objectives

- Understanding the fundamental concepts of Enumerations and Annotations
- Apply the concepts of Generic classes in Java programs
- Demonstrate the fundamental concepts of String operations
- Design and develop web applications using Java servlets and JSP
- 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:
https://nptel.ac.in/courses/106/105/106105191/
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.

11. **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:

12. 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.
13. 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.
14. The students have to answer 5 full questions selecting one full question from each module. 15. 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

M23BCE505D.1	Understanding the fundamental concepts of Enumerations and Annotations
M23BCE505D.2	Apply the concepts of Generic classes in Java programs
M23BCE505D.3	Demonstrate the concepts of String operations in Java
M23BCE505D.4	Develop web based applications using Java servlets and JSP
M23BCE505D.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
M23BCE505D.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE505D.2	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE505D.3	-	-	3	3	-	-	-	-	-	-	-	-	3	3
M23BCE505D.4	-	-	-	3	-	-	-	-	-	-	-	-	3	3
M23BCE505D.5	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCE505D	**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.

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.

5thSemester	**Project Work (PW)** **MINI PROJECT**	**M23BCE506**

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	Complete the final implementation of the project and prepare detailed documentation.
	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
M23BCE506.1	Apply engineering principles to identify, formulate, and solve real-world problems.
M23BCE506.2	Design and develop prototypes or models that address specific engineering challenges.
M23BCE506.3	Collaborate with team members to complete the project successfully.
M23BCE506.4	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
M23BCE506.1	3	3							3			3	3	
M23BCE506.2	3	3	3				3				3	3	3	
M23BCE506.3		3	2	3		3						3	3	
M23BCE506.4				3	3			3		3		3	3	
M23BCE506	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	

7. 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 Course (AE) RESEARCH METHODOLOGY AND INTELLECTUAL PROPERTY RIGHTS	M23BRMK507

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Basic Understanding of Research Concepts**	Before delving into the specifics of engineering research and intellectual property rights, students should have a foundational understanding of what research is, its objectives, and its significance, particularly in the context of engineering.
2	**Familiarity with Ethics in Research**	Basic knowledge of ethics, including common ethical dilemmas and misconduct in research, is essential. This includes understanding issues related to authorship and ethical considerations in the research process.
3	**Literature Review Skills**	Students should have prior experience in conducting literature reviews, including familiarity with bibliographic databases such as Web of Science, Google Scholar, and effective search strategies. This will help them in understanding and analyzing existing knowledge in their research field.
4	**Introduction to Intellectual Property Rights**	A preliminary understanding of intellectual property rights, including patents, copyrights, trademarks, and industrial designs, would be beneficial. This knowledge should include the role of IP in society and basic IP laws, especially in the Indian context.
5	**Technical Reading and Writing Skills**	Competence in reading and comprehending technical documents, including research papers, datasheets, and legal texts, is crucial. Additionally, students should have basic knowledge of how to structure a journal paper and the importance of proper citation and attribution in academic writing.

2. Competencies

S/L	Competency	KSA Description
1	**Understand the research process**	**Knowledge:** Types of research (exploratory, descriptive, explanatory, etc.) Research methodologies (qualitative, quantitative, mixed) Research design (experimental, correlation, causal-comparative) Research ethics principles **Skills:** Identify research problems, Formulate research questions and objectives, Develop research proposals, Conduct literature reviews **Attitudes:** Curiosity and inquisitiveness, Critical thinking and problem-solving, Intellectual honesty and integrity
2	**Apply ethical principles to research**	**Knowledge:** Ethical guidelines for research Ethical issues in research (plagiarism, data fabrication, etc.) Researcher-participant relationships **Skills:** Identify potential ethical dilemmas in research Develop ethical protocols for research Obtain informed consent from participants **Attitudes:** Respect for human subjects, Commitment to research integrity Responsibility for the ethical conduct of research

3		**Knowledge:** Sources of research literature (databases, journals, books) Literature review structure and organization
	Conduct effective literature reviews	Critical appraisal of research articles **Skills:** Search for relevant research literature, Evaluate and synthesize research findings, Organize and present literature review findings **Attitudes:** Persistence and thoroughness, Open-mindedness to different perspectives, Attention to detail
4	**Design research studies**	**Knowledge:** Research designs (experimental, correlational, causal-comparative), Sampling techniques, Data collection methods (surveys, interviews, observations) **Skills:** Develop research instruments, Select appropriate research design, Develop data collection plans **Attitudes:** Creativity and innovation, Flexibility and adaptability, Attention to detail
5	**Understand the concept of intellectual property**	**Knowledge:** Definition and types of intellectual property (patents, copyrights, trademarks, trade secrets, industrial designs) Legal framework for intellectual property protection Economic and social importance of intellectual property **Skills:** Identify intellectual property assets within an organization or project Understand the basics of intellectual property valuation **Attitudes:** Appreciation for the value of intellectual property, Respect for intellectual property rights, Awareness of intellectual property issues in business and research

3. Syllabus

RESEARCH METHODOLOGY & INTELLECTUAL PROPERTY RIGHTS **SEMESTER – V**			
Course Code	**M23BRMK507**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(1:2:0:0)**	SEE Marks	**50**
Total Number of Lecture Hours	**25 hours Theory**	Total Marks	**100**
Credits	**02**	Exam Hours	**03**

Course Objectives:

1. To know the meaning of engineering research.
2. To know the procedure of Literature Review and Technical Reading.
3. To know the fundamentals of patent laws and drafting procedure.
4. To gain awareness of the copyright laws and subject matters of copyrights and designs.
5. To interpret and learn the basic principles of design rights.

Module -1

Introduction: Meaning of research, objectives of engineering research, and motivation in engineering research, types of engineering research, finding and solving a worthwhile problem.

Ethics in engineering research: Ethics in engineering research practice, types of research misconduct, and ethical issues related to authorship.

Module -2

Journal Paper document: structure and approach, Literature Review and Technical Reading: New and existing knowledge in research field, analysis and synthesis of prior art. Bibliographic databases like web of science, Google and Google scholar. Effective search: the way forward, introduction to technical reading conceptualizing research, critical and creative reading, taking notes while reading, reading mathematics and algorithms, reading a datasheet.
Attributions and Citations: Giving credit wherever due, citations: functions and attributes, impact of title and keywords on citations, knowledge flow through citation, styles for citations, citing datasets,

acknowledgments and attributions, what should be acknowledged, acknowledgments in books and dissertations, dedication vs. acknowledgments.

Module -3

Introduction to Intellectual Property (IP): Role of IP in the economic and cultural development of the society, IP governance, IP as a global indicator of innovation, origin of IP, history of IP in India. Major amendments IP laws and acts in India. IP Organizations in India, schemes and programs.
Patents: Conditions for obtaining a patent protection, to patent or not to patent an invention. Rights associated with patents and enforcement of patent rights. Non-patentable matters. Patent infringements and avoiding public disclosure of an invention before patenting.
Process of Patenting: Prior art search, choice of application to be filed, patent application forms, fee structure, types of patent applications. Jurisdiction of filing patent application, publication, pre-grant opposition, examination, and grant of a patent. Validity of patent protection, post-grant opposition, and commercialization of a patent. Need for a patent attorney/ agent. Can a worldwide patent be obtained? Do I need first to file a patent in India? Commonly used terms in patenting, National bodies dealing with patent affairs, utility models.
Case Studies on Patents. Case study of Curcuma (Turmeric) Patent, Case study of Neem Patent, Case study of Basmati patent.

Module -4

Copyrights and Related Rights: Classes of copyrights, criteria for copyright, ownership of copyright, and copyrights of the author. Copyright infringement a criminal offence and cognizable offence. Fair use doctrine. Copyrights and internet. Non-copyright work. Copyright registration. Judicial powers of the registrar of copyrights. Fee structure, copyright symbol, validity of copyright, copyright profile of India. Transfer of copyrights to a publisher. Copyrights and the words 'adaptation', 'Indian work', 'joint authorship', 'publish'. Copyright society, copyright board, and copyright enforcement advisory council (CEAC). International copyright agreements, conventions and treaties.
Case Studies of Copyrights cases: Hawkins Cooker Ltd. vs. Magicook Appliances, KSRTC copyright case.
Trademarks registration: prior art search, eligibility criteria, who can apply for a trademark. Acts and laws. Designation of trademark symbols. Classification of trademarks. Registration of a trademark is not compulsory. Validity of trademark. Types of trademark registered in India. Trademark registry and process for trademarks registration. **Case Studies on Trademarks:** Coca-cola company vs. Bisleri international PVT. Ltd, and Yahoo! Inc. vs. Akash Arora & Anr

Module -5

Industrial Designs: Eligibility criteria, Acts and laws to govern industrial designs. Design rights. Enforcement of design rights. Non-protectable industrial designs India. Protection term. Procedure for registration of industrial designs: Prior art search, application for registration, duration of the registration of a design. Importance of design registration. Cancellation of the registered design. Application forms.
Classification of industrial designs. Designs registration trend in India. International treaties.
Famous case of: Apple inc. vs. Samsung electronics co.
Geographical Indications (GI): acts, laws and rules pertaining to GI. Ownership of GI. Rights granted to the holders. Registered GI in India. Identification of registered GI. Classes of GI. Non-registerable GI. Protection of GI. Collective or certification marks. Enforcement of GI rights. Procedure for GI registration documents required for GI registration. GI ecosystem in India.
Case Studies on GI tags: Case Study of Mysore Silk, Darjeeling Tea, Kancheepuram Silk Sarees, case of Goa's Feni

Text Books:

1. Dipankar Deb • Rajeeb Dey, Valentina E. Balas "Engineering Research Methodology", ISSN 18684394 ISSN 1868-4408 (electronic), Intelligent Systems Reference Library, ISBN 978-981-13- 2946-3 ISBN 978-981-13-2947-0 (eBook), https://doi.org/10.1007/978-981-13-2947-0
2. KOTHARI, C. R. (2004). "Research methodology: Methods and techniques". New age international. 3. Intellectual Property A Primer for Academia by Prof. Rupinder Tewari Ms. Mamta Bhardwa **Reference Book:**
   1. David V. Thiel "Research Methods for Engineers" Cambridge University Press, 978-1-107-03488- 4
   2. Intellectual Property Rights by N.K. Acharya Asia Law House 6th Edition. ISBN: 978-93 81849-30-9

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: Introduction to Research and Intellectual Property	Week 1: Research fundamentals, types of research, research process, ethics Week 2: Intellectual property overview, patents, trademarks Week 3: Copyrights, industrial designs, geographical indications
2	Week 4-6: Literature Review, Research Design, and Data Analysis	Week 4: Literature review, bibliographic databases, citation styles Week 5: Research design, sampling, data collection methods Week 6: Data analysis techniques, research ethics case study
3	Week 7-9: Intellectual Property Law and Enforcement	Week 7: Patent law, patent search, patent drafting Week 8: Trademark law, trademark search, brand management Week 9: Copyright law, fair use, digital copyright
4	Week 10-12: Intellectual Property and Business	Week 10: Intellectual property valuation, licensing, and commercialization Week 11: Intellectual property strategy and management Week 12: Case studies on intellectual property disputes
5	Week 13	Review and Final Exam

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 research methodology 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. 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	Understand the fundamental principles of research methodology.	Research objectives should be clear and based on curiosity. A systematic approach, inspired by the scientific method, ensures transparency and replication. The goal should be to add something new or distinctive, exploring and questioning existing knowledge.
2	Apply ethical considerations in engineering research.	Students as researchers must obtain informed consent from study participants, ensure voluntary participation, protect participant identities and confidentiality, prevent harm to participants, and submit proposals to an institutional review board (IRB) for ethical approval before data collection, ensuring compliance with research objectives and balancing safety with research objectives.
3	Conduct effective literature reviews and technical reading.	Students will learn to conduct a literature review, start by searching for relevant sources, evaluating and selecting them based on quality and relevance, identifying themes, debates, and gaps, outlining your findings logically, and writing your review. Analyze, critique, and compare different sources, highlighting how your research contributes to the ongoing scholarly conversation.
4	Identify and utilize proper attribution and citation styles.	Different disciplines use specific citation styles, such as APA, MLA, or Chicago. In-text citations include author's name and publication year. Reference lists or bibliographies should be compiled at the end of the work. Book citations include author(s), title, publisher, and publication year. Journal article citations include author(s), title, journal name, volume, issue, and publication year.
5	Gain knowledge of different forms of intellectual property (IP) protection.	Patents, copyrights, and trade secrets are legal rights granted by government agencies to inventors, protecting novel processes, machines, and compositions of matter. Trademarks safeguard brand names and symbols, while trade secrets provide confidential information for competitive advantage.
6	Understand the patenting process and its importance. Recognize the significance of copyrights, trademarks, industrial designs, and geographical indications.	A patent is a legal shield granted by a government authority to inventors, providing exclusive rights to an original invention. There are three main types: utility patents, plant patents, design patents, trademarks, copyrights, industrial designs, and geographical indications. Utility patents cover inventions like machines, software, and chemical formulations, while plant patents safeguard unique plant characteristics. Design patents protect product ornamental appearance. Trademarks help build brand identity and prevent confusion.

7	Identify relevant IP organizations and government schemes in India.	The Indian government initiatives include CIPAM, IPRs in School Syllabus, Patent Facilitation Program, National IPR Policy, Technology and Innovation Support Centers, Start-up India, Make in India, National IP Awards, Patent Prosecution Highway, border measures, and support for startups and MSMEs.

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

Course Outcomes (COs)

COs	Description
M23BRMK507.1	**Interpret** the ethical issues in engineering research, including identifying types of research misconduct and evaluating the impact of ethical practices on research outcomes.
M23BRMK507.2	**Analyze literature from diverse bibliographic databases, critically appraise existing research, and synthesize prior art to develop a comprehensive understanding of a chosen research topic.**
M23BRMK507.3	**Apply** appropriate citation styles and techniques, ensuring proper attributions in academic writing to maintain ethical standards and enhance the credibility of research work.
M23BRMK507.4	**Apply the principles of intellectual property rights, including patents, copyrights, and trademarks, to assess the eligibility of an invention or creative work for protection, and navigate the processes for registration and enforcement.**
M23BRMK507.5	**Analyze** the role of intellectual property in economic and cultural development, and explain the historical evolution and contemporary relevance of IP laws and acts, particularly in the Indian context.

CO-PO-PSO Mapping

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

1. **Advanced Research Opportunities:**

- **PhD and Postdoctoral Research: The course equips mechanical engineers with essential research skills, making them strong candidates for advanced studies. This can lead to specialization in emerging fields like computational mechanics, renewable energy systems, and smart manufacturing.**

- **Interdisciplinary Research: Understanding research methodology enables mechanical engineers to collaborate on interdisciplinary projects, combining mechanical engineering with fields like materials science, robotics, and artificial intelligence.**

2. Innovation and Product Development:

- **Patentable Innovations: Knowledge of intellectual property rights allows engineers to protect their innovations, leading to the development of patentable technologies. This is particularly relevant in industries like automotive, aerospace, and manufacturing, where innovation is key to competitiveness.**

- **Start-ups and Entrepreneurship: The course provides a foundation for engineers to start their own ventures, focusing on innovative mechanical products or services. Understanding IP can help secure funding and protect their business ideas.**

3. Career in Research and Development (R&D):

- **Industry R&D Roles: Mechanical engineers with strong research methodology skills are valuable assets in R&D departments. They can lead projects that require rigorous research, data analysis, and the development of new technologies or processes.**

- **Government and Private Research Organizations: Opportunities in organizations like CSIR (Council of Scientific & Industrial Research), DRDO (Defense Research and Development Organization), or private research labs, where engineers can contribute to national and international projects.**

4. Consulting and Advisory Roles:

- **IP Consulting: Engineers with expertise in intellectual property rights can work as consultants, advising companies on patenting strategies, IP management, and innovation protection.**

- **Research Methodology Expert: Mechanical engineers can also serve as advisors or consultants for research projects, helping organizations design and implement robust research methodologies.**

5. Teaching and Academia:

- **Faculty Positions: With advanced knowledge in research methodology and IP, mechanical engineers can pursue teaching careers in universities or technical institutes, contributing to the next generation of engineers.**

- **Curriculum Development: They can also be involved in developing or enhancing engineering curricula, incorporating modern research methods and IP considerations into mechanical engineering programs.**

6. Contribution to Sustainable Development:

- **Innovations for Sustainability: Mechanical engineers can apply their research skills to develop sustainable technologies, focusing on renewable energy, energy efficiency, and reducing the environmental impact of mechanical systems.**

- **Policy Making: With an understanding of the societal impact of engineering solutions, they can contribute to policy-making processes, particularly in areas related to environmental sustainability and technology regulation.**

This course lays the foundation for mechanical engineers to not only excel in their current roles but also to explore new horizons in research, innovation, and entrepreneurship, making a significant impact on their field and society.

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 Theory**	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:

https://sdgs.un.org/goals Video Lectures 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 concems and energy systems.

6. Assessment Details (both CIE and SEE)

Assessment Details (both CIE and SEE)

Continuous internal Examination (CIE)

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
- 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

- 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.
- 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 1 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 legislation and its application.	Students will gain insights into key environmental acts and regulations, and learn how to apply them in real-world scenarios.

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	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
M23BESK508.1												
M23BESK508.2												

M23BESK508.3												
M23BESK508.4												
M23BESK508.5												
M23BESK508												

9. Assessment Plan

Continuous Internal Evaluation (CIE)

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

Semester End Examination (SEE)

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

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	-	SEEMarks	-
Credits	**0**	TotalMarks	100
Activities Report Evaluation by College NSS Officer at the end of every semester (3rd to 6th semester)			

Courseobjectives:

National Service Scheme (NSS) willenable studentsto:

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, 5 R'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 actionable business proposal for enhancing the village income 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, AtmanirbharBharath, Make in India, Mudra scheme, Skill development programs etc.
9. Spreading public awareness under rural outreach programs.(minimum5 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:

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

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

Sem	Topics / Activities to be Covered

3rd Sem for 25 Marks	Organic farming, Indian Agriculture (Past, Present, and Future) Connectivity for marketing. Waste management– Public, Private and Govt organization, 5 R's. Setting of the information imparting club for women leading to contribution in social andeconomic 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 forimplementation. 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. Foreg. Digital India, Skill India, Swachh Bharat, AtmanirbharBharath, Make in India, Mudra scheme, Skill developmentprograms etc. Spreading public awareness under rural outreach programs.(minimum5 programs). Social connect and responsibilities.
6thSem for 25 Marks	Plantation and adoption of plants. Know your plants. Organize National integration and social harmony events /workshops /seminars. (Minimum 02programs). 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:

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

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

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

7.	Developing Sustainable Water management system for rural areas and implementation approaches.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Site selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
8.	Contribution to any national-level initiative of the Government of India. For eg. Digital India, Skill India, Swachh Bharat,Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
9.	Spreading public awareness under rural outreach programs.(minimum5 programs). ///// Social connect and responsibilities.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
10.	Plantation and adoption of plants. Know your plants.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/ Government Schemes officers / campus etc…	Place selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSSofficer
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 NSSofficer

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 NSSofficer

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.	**Sectorwise 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 accordingto the scheme and syllabus.
- At the end of every semester student performance has to be evaluated by the NSS officer for theassigned activity progress and its completion.
- At last in 6thsemester consolidated report of all activities from 3rdto 6thsemester, compiledreport 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 by each 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 semesterfor CIE entry in the VTU portal.
25 marks CIE entry will be entered in University IA marks portal at the end of each semester 3 rdto 6thsem, 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.

Semester - V			
PHYSICAL EDUCATION (SPORTS & ATHLETICS) — I			
Course Code	**M23BPEK509**	CIE Marks	**100**
Number of Lecture Hours/Week(L: T: P: S)		SEEMarks	
Total Number of Lecture Hours		TotalMarks	100
Credits	**0**	Exam Hours	**-**

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

CO1. Understand the fundamental concepts and skills of Physical Education, Health, Nutrition and Fitness.

CO2. Familiarization of health-related Exercises, Sports for overall growth and development.

CO3. Create a foundation for the professionals in Physical Education and Sports.

CO4. Participate in the competition at regional/state / national / international levels.

CO5. Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Module-1

Orientation **(5 hours)**

A. Lifestyle

B. Fitness

C. Food & Nutrition

D. Health & Wellness

E. Pre-Fitness test.

Module-2

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

A. Warmingup(FreeHandexercises)
B. Strength—Push-up/Pull-ups
C. Speed—30MtrDash
D. Agility —ShuttleRun
E. Flexibility—SitandReach
F. CardiovascularEndurance—HarvardstepTest

Module-3

Recreational Activities: **(10 hours)**

A. Posturaldeformities.
B. Stressmanagement.
C. Aerobics.
D. Traditional Games.

Scheme and Assessment for auditing the course and Grades:

Sl. No.	Activity	Marks
1.	Participationofstudentinallthemodules	20
2.	Quizzes—2, eachof15marks	30
3.	Finalpresentation/exhibition/Participationin competitions/practicalon specific tasksassignedtothestudents	50
	Total	100

Semester - IV

PHYSICAL EDUCATION (SPORTS & ATHLETICS) — II

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

CO1. Understand the ethics and moral values in sports and athletics
CO2. Perform in the selected sports or athletics of the student's choice.
CO3. Understandtheroles andresponsibilities of organisation and administration of sports and games.

Module-1

EthicsandMoralValues (5 hours)

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

Module-2

Specific Games (Any one to be selected by the student) **(20 hours)**
A. Volleyball — Attack, Block, Service, Upper Hand Pass and Lower hand Pass.
B. Throwball — Service, Receive, Spin attack, Net Drop & Jump throw.
C. Kabaddi — Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus.
D. Kho-Kho — Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 Up.
E. Table Tennis — Service (Fore Hand & Back Hand), Receive (Fore Hand & Back Hand), Smash.
F. Athletics (Track / Field Events) — Any event as per availability of Ground.
Module-3
Role of Organisation and administration **(5 hours)**

Scheme and Assessment for auditing the course and Grades:

Sl. No.	Activity	Marks
1.	Participationofstudentinallthemodules	20
2.	Quizzes—2, each of 15marks	30
3.	Final presentation/exhibition/Participationin competitions/practicalon specific task sassigned to the students.	50
	Total	100

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		SEEMarks	**-**
Credits	**0**	TotalMarks	**100**
Evaluation Method: Objective type Theory / Practical / Viva-Voce			

Courseobjectives:

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 several ailments such as

- coronary heart disease,
- depression,
- anxiety disorders, • asthma, and
- Extensive rehabilitation for disorders including musculoskeletal problems andtraumatic brain injury.

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

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

- Physical

1. Improved body flexibility and balance
2. Improved cardiovascular endurance (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 III

- Yoga, its origin, history and development. Yoga, its meaning, definitions.

- Different schools of yoga, Aim and Objectives of yoga, importance of prayer
- Yogic practices for a common man to promote positive Health
- Rules to be followed during yogic practices by the practitioner
- Yoga its misconceptions,
- Difference between yogic and non-yogic practices
- Surya namaskar prayer and its meaning, Need, importance and benefits of Surya namaskar12count, 2 rounds
- Asana, Need, importance of Asana. Different types of asanas. Asana its meaning by name,technique, precautionary measures and benefits of each asana
- Different types of Asanas

a. Sitting
1. Padmasana
2. Vajrasana

b. Standing
1. Vrikshana
2. Trikonasana

c. Prone line
1. Bhujangasana
2. Shalabhasana

d. Supine line
1. Utthitadvipadasana
2. Ardhahalasana

Semester IV

- Patanjali'sAshtanga Yoga, its need and importance.
- Yama :Ahimsa, satya, asteya, brahmacarya, aparigraha.
- Niyama :shoucha, santosh, tapa, svaadhyaya, Eshvarapranidhan
- Suryanamaskar12 count- 4 rounds of practice
- Asana, Need, importance of Asana. Different types of asana. Asana its meaning by name, technique, precautionary measures and benefits of each asana.
- Different types of Asanas

a. Sitting
1. Sukhasana
2. Paschimottanasana

b. Standing
1. ArdhakatiChakrasana
2. ParshvaChakrasana

c. Prone line
1. Dhanurasana

d. Supine line
1. Halasana
2. Karna Peedasana

- Meaning, importance and benefits of Kapalabhati.
- 40 strokes/min 3 rounds
- Meaning, Need, importance of Pranayama. Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama.
- Pranayama

1. Suryanuloma –Viloma
2. Chandranuloma-Viloma
3. Suryabhedana

4. Chandra Bhedana
5. Nadishodhana

Semester V

- Patanjali'sAshtanga Yoga its need and importance.
- Ashtanga Yoga

1. Asana
2. Pranayama
3. Pratyahara

- Asana its meaning by name, technique, precautionary measures and benefits of each asana
- Different types of Asanas

a. Sitting 1. ArdhaUshtrasana 2. Vakrasana 3. Yogamudra in Padmasana
b. Standing 1. UrdhvaHastothanasana 2. Hastapadasana 3. ParivrittaTrikonasana 4. Utkatasana
c. Prone line 1. PadangushthaDhanurasana 2. PoornaBhujangasana / 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

Semester VI

- Ashtanga Yoga

1. Dharana
2. Dhyana (Meditation)
3. Samadhi

- Asana by name, technique, precautionary measures and benefits of each asana
- Different types of Asanas

a. Sitting 1. Bakasana 2. Hanumanasana 3. EkapadaRajakapotasana 4. Yogamudra in Vajrasana b. Standing 1. Vatayanasana 2. Garudasana

c. Balancing 1. Veerabhadrasana 2. Sheershasana
d. Supine line 1. Sarvangasana 2. SetubandhaSarvangasana 3. Shavasanaa (Relaxation poisture).

- Revision of Kapalabhati practice 80 strokes/min - 3 rounds
- Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama 1. Bhastrika 2. Bhramari
- Meaning, Need, importance of Shatkriya.
- Different types. Meaning by name, technique, precautionary measures and benefits of each Kriya 1. Jalaneti&sutraneti 2. Nouli (only for men) 3. SheetkarmaKapalabhati

Course outcomes (Course Skill Set):

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

- Understand the meaning, aim and objectives of Yoga.
- Perform Suryanamaskar and able to Teach its benefits.
- Understand and teach different Asanas by name, its importance, methods and benefits.
- Instruct Kapalabhati and its need and importance.
- Teach different types of Pranayama by its name, precautions, procedure and uses
- Coach different types of Kriyas , method to follow and usefulness.

Assessment Details (both CIE and SEE)

- Students will be assessed with internal test by a. Multiple choice questions b. Descriptive type questions (Two internal 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 YaminiMuthanna

Web links and Video Lectures (e-Resources):

Refer links

1. https://youtu.be/KB-TYlgd1wE
2. 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	**Python Proficiency**	**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	**Django Framework Mastery**	**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	**Database Management**	**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	**Front-End Development Skills**	**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	**Security Best Practices**	**Knowledge**: Techniques for troubleshooting and debugging. **Skills**: Implement solutions efficiently and evaluate their effectiveness **Attitudes:** Analyze complex issues and propose effective solutions.
6	**Testing and Debugging**	**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.

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 hrs Theory +20 hrs 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 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.
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 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 andhow 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 preclass videos on Django's ORM, and use class time to discuss advanced queries and perform hands-on exercises.
6	Continuous	Use formative assessments (quizzes, coding challenges) and provide regular
	Assessment and Feedback	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) Continuous Internal Evaluation:

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

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

FinalCIE Marks =(A) + (B) Semester

End Examination pattern:

1. Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
2. There shall be 2 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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
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						**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						**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.

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 Learningand 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, performs cross-validation, and select the best model for a given task. **Attitudes:** Objectivity in model assessment.
5	Ethical Considerations inMachine 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 Theory**	Total Marks	**100**

Credits	**03**	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(8 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 (8 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 (8 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 (6 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. Introduction to Machine Learning, Fourth Edition By EthemAlpaydin.

VIDEO LINKS:

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

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 scikitlearn) 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
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

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

Semester End Examination:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module.
- Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	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	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
1	2												3	3
2		3											3	
3			3		3									3
4				3	3								3	
5				3						3				3
Avg	**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.

Job Prospects:

1. **Thriving in a High-Demand Field:** ML expertise is highly sought after across industries. This course equips you with in-demand skills, paving the way for careers as Machine Learning Engineers, Data Scientists, AI Specialists, and more.
2. **Solving Real-World Challenges:** From optimizing manufacturing processes and developing intelligent systems to revolutionizing healthcare and transportation, ML skills are driving innovation. Your expertise will be invaluable in tackling these challenges.
3. **Future-Proofing Your Career**: ML is constantly evolving, and a strong foundation ensures you can adapt to new advancements. This course equips you with the tools and mindset to stay ahead of the curve and thrive in a dynamic job market.

6th Semester	Professional Elective (PE) BLOCKCHAIN TECHNOLOGY	M23BCE603A

1. Prerequisites

S/L	Proficiency	Prerequisites
1.	Basic Programming Skills	Understanding of programming concepts andknowledge 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 principlesis 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:** Understandthe 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.	**BasicKnowledge 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. Understandingand 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**: Understandthe 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 **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 Programming concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with realworld competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. **Assessment Details (both CIE and SEE) Continuous Internal Evaluation:**

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

FinalCIE 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
M23BCE603A.1	Explain fundamentals of blockchain and how it works.
M23BCE603A.2	Implement security measures through cryptographic principles.
M23BCE603A.3	Describe fundamentals of cryptocurrencies and their role in economics.
M23BCE603A.4	Analyse and demonstrate the Ethereum.
M23BCE603A.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
M23BCE603A.1	3												3	3
M23BCE603A.2		3											3	3
M23BCE603A.3			3										3	3

M23BCE603A.4	**3**												**3**	**3**
M23BCE603A.5	**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. Blockchain has the potential to drive significant social and economic changes, promoting financial inclusion and transparency.

6thSemester	PROFESSIONAL ELECTIVE – II (PE) COMPUTER VISION	M23BCE603B

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Mathematics**	**Linear Algebra:** Understanding of vectors, matrices, eigenvalues, and eigenvectors, which are fundamental in image transformations, camera models, and feature detection. **Calculus:** Proficiency in differential calculus for understanding optimization in machine learning models, as well as the image brightness constancy equation. **Probability and Statistics:** Basic concepts to understand statistical filtering, robust estimation, and error modelling.
2	**Basic Algorithm Design**	Familiarity with algorithm design principles, data structures (e.g., arrays, lists, trees, graphs), and complexity analysis.
3	**Programming**	Familiarity with Python, including libraries like NumPy, OpenCV, and Matplotlib for image processing tasks.
4	**Fourier Transforms**	Understanding of Fourier transforms and their applications in signal processing.
5	**Basic Image Processing**	Prior exposure to image processing techniques such as filtering, convolution, and Fourier transforms. This will help students grasp the more advanced concepts covered in the course

2. Competencies

S/L	Competency	KSA Description
1	**Overview of Computer Vision and Its Applications**	**Knowledge:** Understanding the fundamentals of imaging geometry, radiometry, digitization, camera models, and projection techniques. **Skills:** Ability to apply concepts from linear algebra and calculus to understand imaging geometry and transformations. **Attitudes:** Eagerness to explore the foundational aspects of computer vision and understand the underlying principles of image formation and processing.
2	**Feature Detection and Segmentation**	**Knowledge:** Feature Detection: Understanding of edge, corner, line, and curve detection techniques, along with descriptors like SIFT, HOG, and shape context. Segmentation Techniques: Familiarity with various segmentation methods such as active contours, split & merge, watershed, graph-based segmentation, and normalized cuts. **Skills:** Algorithm Development: Ability to implement and modify feature detection algorithms for specific tasks in computer vision. **Attitudes:** Precision: A meticulous approach to identifying and extracting features and segmenting images accurately.
3	**Camera Calibration**	**Knowledge:** In-depth understanding of camera models, including intrinsic and extrinsic parameters, radial lens distortion, and various projection techniques. **Skills:** Skill in calibrating cameras using calibration techniques and understanding the impact of different camera models on image acquisition. **Attitudes:** A critical approach to understanding and applying camera calibration techniques, ensuring accurate image acquisition.

4	**Motion Representation**	**Knowledge:** Understanding the motion field of rigid objects, motion parallax, optical flow, affine flow, and the image brightness constancy equation. **Skills: Motion Analysis:** Understanding the motion field of rigid objects, motion parallax, optical flow, affine flow, and the image brightness constancy equation. **Attitudes:** A proactive approach to tackling challenges in motion representation and finding innovative solutions.
5	**Motion Tracking and**	**Knowledge:** Understanding of statistical filtering, iterated estimation, Kalman filters, and their applications in motion tracking.
	Object Recognition	**Skills:** Ability to develop and implement motion tracking algorithms using statistical filtering and estimation techniques. **Attitudes:** A commitment to achieving high accuracy in motion tracking and object recognition tasks.

3. Syllabus

COMPUTER VISION SEMESTER – III			
Course Code	**M23BCE603B**	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**

Course Objectives:

- Learn how digital images are formed, represented, and processed using techniques like filtering and transformations.
- Understand and apply methods to identify edges, corners, and other features, as well as divide images into meaningful regions.
- Acquire skills to set up cameras accurately and use their parameters to rebuild three-dimensional scenes from images.
- Study techniques to represent movement in images and implement methods to follow moving objects over time.
- Explore and apply approaches to identify objects within images and describe their shapes effectively.

Module -1

Overview of computer vision and its applications: Image Formation and Representation: Imaging geometry, radiometry, digitization, cameras and Projections, rigid and affine transformation

Image Processing: Pixel transforms, colour transforms, histogram processing, histogram equalization, filtering, convolution, Fourier transformation and its applications in sharpening, blurring and noise removal

Module -2

Feature detection: edge detection, corner detection, line and curve detection, active contours, SIFT and HOG descriptors, shape context descriptors, Morphological operations

Segmentation: Active contours, split & merge, watershed, region splitting, region merging, graph-based segmentation, mean shift and model finding, Normalized cut

Module -3

Camera calibration: camera models; intrinsic and extrinsic parameters; radial lens distortion; direct parameter calibration; camera parameters from projection matrices; orthographic, weak perspective, affine, and perspective camera models.

Module -4

Motion representation: the motion field of rigid objects; motion parallax; optical flow, the image brightness constancy equation, affine flow; differential techniques; feature-based techniques; regularization and robust estimation

Module -5
Motion tracking: statistical filtering; iterated estimation; observability and linear systems; the Kalman filter. **Object recognition and shape representation:** alignment, appearance-based methods, invariants, image eigenspaces.
Text Books: 1.Computer Vision: Algorithms and Applications, R. Szeliski, Springer, 2011. 2. Introductory techniques for 3D computer vision, E. Trucco and A. Verri, Prentice Hall, 1998. **Reference Books:** 1. Computer Vision: Algorithms and Applications, Richard Szeliski 2. Multiple View Geometry in Computer Vision , Richard Hartley and Andrew Zisserman

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3:	**Overview of computer vision and its applications:** Image Formation and Representation: Imaging geometry, radiometry, digitization, cameras and Projections, rigid and affine transformation **Image Processing:** Pixel transforms, colour transforms, histogram processing, histogram equalization, filtering, convolution, Fourier transformation and its applications in sharpening, blurring and noise removal
2	Week 4-6:	**Feature detection:** edge detection, corner detection, line and curve detection, active contours, SIFT and HOG descriptors, shape context descriptors, Morphological operations **Segmentation:** Active contours, split & merge, watershed, region splitting, region merging, graph-based segmentation, mean shift and model finding, Normalized cut
3	Week 8-11:	**Camera calibration:** camera models; intrinsic and extrinsic parameters; radial lens distortion; direct parameter calibration; camera parameters from projection matrices; orthographic, weak perspective, affine, and perspective camera models.
4	Week 7-8:	**Motion representation:** the motion field of rigid objects; motion parallax; optical flow, the image brightness constancy equation, affine flow; differential techniques; feature-based techniques; regularization and robust estimation
5	Week 9-12:	**Motion tracking:** statistical filtering; iterated estimation; observability and linear systems; the Kalman filter. **Object recognition and shape representation:** alignment, appearance-based methods, invariants, image eigenspaces.

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Interactive Lectures	Use lectures to introduce key concepts in computer vision, image processing, and motion analysis. Ensure that each lecture includes visual aids, diagrams, and real-world examples to illustrate complex ideas.
2	Hands-on Labs and Workshops	Conduct lab sessions where students can apply theoretical knowledge by coding and experimenting with image processing algorithms, feature detection techniques, and camera calibration tasks using Python and relevant libraries.
3	Collaborative Learning	Group Projects: Assign group projects where students work together to solve complex computer vision problems, encouraging peer-to-peer learning and collaboration.
4	Problem-Based Learning	Real-World Case Studies: Introduce problem-based learning by presenting students with real-world case studies that require them to apply computer vision techniques to solve practical problems.

6. Assessment Details (both CIE and SEE) Continuous Internal Evaluation:

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 Professional Elective Course (PE)

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 Examinations:

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

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module. • Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understand Image Formation and Processing Techniques	Students will be able to explain the principles of image formation, representation, and processing, including key techniques like filtering, convolution, and Fourier transformation.
2	Apply Feature Detection and Segmentation Methods	Students will demonstrate the ability to implement and apply various feature detection and image segmentation methods to extract and analyse meaningful information from images.
3	Calibrate Cameras and Perform 3D Reconstruction	Students will acquire the skills to calibrate cameras, understand intrinsic and extrinsic parameters, and use projection techniques for accurate 3D scene reconstruction.
4	Analyse and Represent Motion in Images	Students will learn to analyse and represent motion in images and videos using techniques like optical flow, motion parallax, and differential methods, understanding the principles behind motion tracking.
5	Implement Motion Tracking and Object Recognition Systems	Students will be able to develop and implement systems for motion tracking and object recognition using statistical filtering, Kalman filters, and appearance-based methods.
6	Integrate Theoretical Knowledge with Practical Applications	Students will integrate their understanding of computer vision concepts with practical applications, solving real-world problems through hands-on projects and case studies.

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

Course Outcomes (COs)

COs	Description
M23BCE603B.1	Apply image processing techniques like filtering and convolution to enhance and modify digital images.
M23BCE603B.2	Analyse and evaluate feature detection and segmentation methods to extract important features and divide images into meaningful regions.
M23BCE603B.3	Calibrate cameras and reconstruct 3D scenes from 2D images using projection techniques.
M23BCE603B.4	Evaluate the effectiveness of motion analysis and tracking systems by critically analysing the application of techniques like optical flow and motion parallax in various scenarios.
M23BCE603B.5	Design and create object recognition systems that accurately identify and classify objects in different environments.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCE603B.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE603B.2	3	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE603B.3	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE603B.4	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCE603B.5	**-**	**-**		3	3	-	-	-	-	-	-	-	3	3
M23BCE603B	**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:

Revolutionizing Autonomous Vehicles:
As self-driving technology advances, computer vision specialists will play a critical role in developing and refining the visual systems that enable vehicles to navigate roads safely and efficiently, making autonomous transportation more reliable and widespread.

Transforming Healthcare with Advanced Diagnostics:
Computer vision will drive innovations in medical imaging, enabling more accurate and automated diagnoses through techniques like automated tumour detection and real-time monitoring of patient conditions, leading to improved patient outcomes and personalized treatments.

Enhancing Augmented and Virtual Reality Experiences:
Experts in computer vision will contribute to creating more immersive and interactive AR and VR environments by improving object recognition, gesture tracking, and environmental mapping, enriching experiences in gaming, training, and education.

Advancing Security and Surveillance Technologies:
Computer vision will enhance security systems through improved facial recognition, anomaly detection, and automated monitoring, leading to more effective crime prevention and public safety measures.

Innovating Robotics and Automation:
The integration of computer vision into robotics will lead to smarter, more capable robots that can perform complex tasks in manufacturing, healthcare, and service industries, driving automation and increasing efficiency across various sectors.

Developing Smart Cities and IoT Solutions:
Computer vision will be pivotal in building smart cities by providing advanced solutions for traffic management, environmental monitoring, and urban planning, leading to more connected, efficient, and sustainable urban environments.

6th Semester	Professional Elective- II COMPILER DESIGN	M23BCE603C

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Proficiency in at least one programming language (e.g., C, C++, Java) is essential. This includes understanding data structures (like arrays, linked lists, trees), algorithms, and basic programming concepts.
2	**Basic Knowledge of Linear Algebra**	Understanding of vectors, matrices, matrix multiplication, and other fundamental linear algebra concepts.
3	**Formal Languages and Automata Theory**	Knowledge of finite automata, regular expressions, context-free grammars, and pushdown automata.
4	**Syntax and Semantics**	Understanding of syntax (how code is structured) and semantics (the meaning of code) of programming languages.
5	**Operating Systems**	Concepts such as process management, memory management, and file systems can be beneficial.
6	**Modular Design**	Knowledge of designing software in a modular and maintainable way, which is crucial for building complex systems like compilers.

2. Competencies

S/L	Competency	KSA Description
1	**Introduction:**	**Knowledge:** Understanding why compilers are needed and what they do (e.g., translating high-level code to machine code). **Skills:** Skills in analyzing and designing algorithms and data structures for tasks such as parsing and optimization. Familiarity with tools like Lex (for lexical analysis) and Yacc/Bison (for parsing). **Attitudes:** Ability to adapt to new methods, tools, and languages as you dive deeper into compiler design. Recognizing that compiler design can be intricate and requires thorough testing and debugging.
2	**Lexical Analysis:**	**Knowledge:** Understanding what lexical analysis is and why it's essential for compiler design. It involves scanning the input source code and breaking it into tokens. Knowledge of various types of tokens (e.g., keywords, operators, identifiers) and their typical patterns in programming languages. **Skills:** Ability to write code for lexical analysis, including implementing tokenization logic and using tools like Lex or Flex. Ability to implement mechanisms for detecting and reporting lexical errors, including providing meaningful error messages. **Attitudes:** Approach problems methodically, breaking down the task of tokenization into manageable components. Willingness to iteratively test and refine your lexical analyzer to handle all expected and edge-case inputs.

3	**Syntax Analysis:**	**Knowledge:** Understanding the role of syntax analysis in translating source code into a parse tree or abstract syntax tree (AST), and its importance in ensuring code conforms to grammatical rules. **Skills:** Ability to write and implement parsers using different techniques (e.g., recursive descent, table-driven parsers). Skills in designing meaningful error messages and handling syntax errors gracefully. **Attitudes:** Recognizing the importance of precise grammar definitions and the potential impact of small errors in syntax rules. Ability to adapt to different programming languages and their specific syntactic
		rules when designing or using parsers.
4	**Syntax directed translation:**	**Knowledge:** Familiarity with the concept of attributed grammars, where grammar rules are augmented with attributes and semantic functions. Knowledge of how intermediate representations are translated into target code and the role of syntax-directed translation in this process. **Skills:** Proficiency with tools and frameworks that support syntax-directed translation, such as parser generators (Yacc/Bison, ANTLR) and compiler construction kits. **Attitudes:** Ability to approach translation problems methodically, breaking down complex syntax and semantic issues into manageable components.
5	**Intermediate Code Generation:**	**Knowledge:** Knowledge of different forms of IR, such as abstract syntax trees (ASTs), three-address code (TAC), and intermediate languages (e.g., LLVM IR). Understanding of control flow graphs (CFGs) and their role in representing the flow of control within a program. **Skills:** Ability to implement code that generates intermediate representations from high-level source code. Skills in selecting appropriate low-level instructions and mapping intermediate code to these instructions. **Attitudes:** Recognizing the importance of precise IR generation and optimization to ensure correct and efficient target code. Keeping up with new methods and best practices in intermediate code generation and optimization.

3. Syllabus

COMPILER DESIGN SEMESTER – VI			
Course Code	**M23BCE603C**	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**

Course Objectives:

- To teach concepts of language translation and phases of compiler design.
- To describe the common forms of parsers.
- To inculcate knowledge of parser by parsing LL parser and LR parser.
- To demonstrate intermediate code using technique of syntax directed translation.

Module -1

Introduction : Language processors; The structure of a Compiler; The evolution programming languages; The science of building a Compiler; Applications of compiler technology; Programming language basics.

Module -2

Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.lexical analyzer generator, Finite automate.
Module -3
Syntax Analysis: Introduction, Role Of Parsers, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers
Module -4
Syntax-Directed Translation: Syntax-directed definitions; Evaluation orders for SDDs; Applications of syntax-directed translation; Syntax-directed translation schemes.
Module -5
Intermediate Code Generation: Variants of syntax trees; Three-address code; Translation of expressions; Control flow; Back patching; Switch statements; Procedure calls.
TEXTBOOKS: 1. Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007
2. Dhamdhere, D. M., "Compiler Construction Principles and Practice", 2nd edition, Macmillan India Ltd., New Delhi, 2008 **Reference Books:** 1.Kenneth C. Louden, "Compiler Construction: Principles and Practice", Thompson Learning, 2003

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1-3: **Introduction**	Language processors; The structure of a Compiler; The evolution programming languages; The science of building a Compiler; Applications of compiler technology; Programming language basics.
2	Week 4-6: **Lexical analysis**	The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.lexical analyzer generator, Finite automate.
3	Week 7-8: **Syntax Analysis**	Introduction, Role Of Parsers, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers
4	Week 9-11: **Syntax-Directed Translation**	Syntax-directed definitions; Evaluation orders for SDDs; Applications of syntaxdirected translation; Syntax-directed translation schemes.
5	Week 12-13 : **Intermediate Code Generation**	Variants of syntax trees; Three-address code; Translation of expressions; Control flow; Back patching; Switch statements; Procedure calls.

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	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.

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:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module.
- Marks scored will be proportionally scaled down to 50 marks **7. Learning Objectives**

S/L	Learning Objectives	Description
1	Introduction	Gain a clear understanding of what a compiler is, why it is needed, and how it fits into the process of translating high-level source code into executable machine code.
2	Lexical analysis	Gain practical skills in implementing lexical analyzers to recognize and process tokens from source code.
3	Syntax analysis	Grasp the significance of syntax analysis in the compilation process, including its role in ensuring that source code adheres to the grammatical rules of the programming language.
4	Syntax directed translation	Understand how syntax-directed definitions and attributed grammars are used to associate semantic rules with syntax rules.
5	Intermediate code generation	Learn how intermediate code is generated from the syntax-directed translation process and its role in the compilation pipeline..

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

Course Outcomes (COs)

Cos	Description
M23BCE603C.1	Apply knowledge of different phases and passes of the compiler
M23BCE603C.2	Analyse formal languages for creating regular expressions to define token patterns
M23BCE603C.3	Construct the intermediate representation considering the type systems
M23BCE603C.4	Design and implement different types of parsers i.e. Top-Down and Bottom-up parsers and construct LL, SLR, CLR, and LALR parsing table.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2

M23BCE603C.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE603C.2	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE603C.3	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCE603C.4	-	-	-	3	-	-	-	-	-	-	-	-	3	3
M23BCE603C	3	3	3	3	-	-	-	-	-	-	-	-	3	3

9. Assessment Plan

Continuous Internal Evaluation (CIE)

	CO1	**CO2**	**CO3**	**CO4**	**Total**
Module 1	**6**				**6**
Module 2	**6**	**6**			**12**
Module 3		**7**	**6**		**13**
Module 4			**6**	**7**	**13**
Module 5				**6**	**6**
Total	**12**	**13**	**12**	**13**	**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. Advanced Optimization Techniques

- **Machine Learning and AI**: Leveraging machine learning and artificial intelligence to improve optimization strategies, such as predicting optimal code transformations or automating performance tuning.
- **Adaptive Optimization**: Developing compilers that adapt to runtime information and dynamically optimize code based on real-time performance metrics. **2. Integration with Modern Hardware Architectures**
- **Heterogeneous Computing**: Designing compilers to efficiently target heterogeneous computing environments, including CPUs, GPUs, TPUs, and other accelerators.
- **Customizable Architectures**: Supporting new and customizable hardware architectures, such as domainspecific processors and reconfigurable hardware (FPGAs).

3. Support for New Programming Paradigms

- **Parallel and Concurrent Programming**: Enhancing compilers to better support parallelism and concurrency, including automatic parallelization and synchronization.
- **Domain-Specific Languages (DSLs)**: Creating compilers that efficiently handle DSLs tailored for specific applications, such as data science, machine learning, or graphics.

4. Increased Focus on Security

- **Security-Oriented Optimizations**: Incorporating security considerations into optimization strategies to mitigate vulnerabilities such as buffer overflows, side-channel attacks, and other exploits.
- **Secure Compilation Practices**: Developing techniques for producing secure code, including tools for detecting and mitigating vulnerabilities during the compilation process.

6thSemester	Professional Elective -II (PE) SOCIAL NETWORK ANALYSIS	M23BCE603D

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 media, 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 isualizations **Attitudes:** Analytical mindset for interpreting complex network structures Curiosity about social dynamics and their digital representations
3	**Extraction and mining communities in web social networks**	**Knowledge:** Definitions ofsocial 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. **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	**M23BCE603D**	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**

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.
   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,CascadeBehaviours 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 NetworksMalicious 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 TanmoyChakraborty , Publisher. Wiley · Publication date. 1 October 2021 2. .GuandongXu ,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 Passantand 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	Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web – Statistical Properties of Social Networks Network analysis - Development of Social Network Analysis - Key concepts and measures in network analysis - Discussion networks - Blogs and online communities - Web-based networks.
2	Week 4-6: 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 - Hybrid Representations - Modelling and aggregating social network data – Random Walks and their Applications –Use of Hadoop and Map Reduce - Ontological representation of social individuals and relationships.
3	Week 8-11: EXTRACTION AND MINING COMMUNITIES IN WEB SOCIAL NETWORKS	Network Extracting evolution of Web Community from a Series of Web Archive – Detecting communities in 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 – Decentralized online social networks – Multi-Relational characterization of dynamic social network communities.
4	Week 7-8: 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 – 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	Week 9-12: APPLICATIONS	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, Explaining Scientific and Technical Emergence Forecasting, Social Network Analysis for Biometric Template Protection

6. 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	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
	TotalMarks			**50**	**20**

FinalCIE Marks =(A) + (B)

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

Semester End Examination:

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module. • Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Learning Objectives	Description
1	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	Understanding and predicting human behaviour for social communities – User data management – Inference and Distribution – Enabling new human experiences –
	privacy issues	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
M23BCE603D.1	**Understand**Semantic and Social Web developments, **apply** statistical properties and key measures in social networks and network analysis techniques to online communities and discussion networks.
M23BCE603D.2	**Apply** visualizing tools on online social networks for modelling and aggregating data.
M23BCE603D.3	**Analyse** and evaluate web and social network communities, applying detection methods and exploring decentralized and multi-relational dynamics.
M23BCE603D.4	**Analyse** human behaviour prediction, user data management, privacy, trust, security, emotion classification, opinion assessment, and biometric protection in social networks.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCE603D.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCE603D.2	3		-	-	2	-	-	-	-	-	-	-	3	-
M23BCE603D.3	-	3	-	-	-	-	-	-	-	-	-	-	-	2
M23BCE603D.4	-	3	-	-	2	-	-	-	-	-	-	-	-	2
M23BCE603D	**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 multidimensional 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.
- 8.**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	M23BCE604A

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**	Understandingthe 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	KSADescription
1	**Core Java Concepts**	**Knowledge:** Proficiency with Java syntax, including how to write correct and efficient code using Java's syntax rules. **Skills:** Learningthebasicconceptsof 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:** Practicingtheconcept offunctionalprogramming **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 eco system, 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	**M23BCE604A**	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**

Course objectives: This course will enable students to
Course Objective:
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**	**Competency:** It involves understanding its fundamental aspects, core features. **Knowledge:** The knowledge of basic concepts such as variables, datatypes, programming structure and execution environment. **Skills:** Acquiring the knowledge of basic concepts of Java programming.
2	Week 3-4: **Operators**	**Competency**: Understanding the usage and applying various types of operators effectively. **Knowledge:** It involves understanding and applying different types of operators. **Skills:** Usage of different operators to perform specific operations.
3	Week 5-6: **Introducing Classes**	**Competency**: Understanding the design, implement, and utilize classes effectively **Knowledge:** Encompasses the understanding of classes concept in programming. **Skills:** Ability to use the concept of classes and object to solve related problems in Java.

4	Week 7-8: **Packages and Interfaces**	**Competency**: It involves understanding purpose, usage, and interaction of packages and interfaces within Java applications. **Knowledge:** Knowledge of packages and interfaces. **Skills:** 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**	**Competency**: Involves understanding and effectively using basic concepts of Applets and string methods. **Knowledge:** Understanding the Applet viewer and different string methods to perform specific operations. **Skills:** 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 aidslike videos/animations to enhance understanding of concepts.
3	**Collaborative Learning**	Encourage collaborative learning for improved competency application.
4	**HigherOrder Thinking (HOTS) Questions:**	Pose HOTS questions to stimulate critical thinking related to each competency.
5	**Problem-Based Learning(PBL)**	Implement PBL to enhance analytical skills and practical application of competencies
6	**Pair Programming**	Incorporate pair programming sessions where students collaborate in pair sto solve coding ask so rwork 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: Professional Elective Course (PE)
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 Course (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

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

7. Learning Objectives

S/L	Learning Objectives	Description
1	**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
M23BCE604A.1	Understanding the basic concepts of java to solve real time problems.
M23BCE604A.2	Apply the object-oriented concepts of java and exception handling concepts to implement java program.
M23BCE604A.3	Analyze I/O and String handling concept to develop an application program.
M23BCE604A.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	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
M23BCE604A.1	3													
M23BCE604A.2		3												
M23BCE604A.3			3										3	3
M23BCE604A.4				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						**50**

Semester End Examination (SEE)

	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						**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 –I INTRODUCTION TO DATA STRUCTURES	M23BCE604B

3. 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	KSADescription

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. **Use of Pointers and Dynamic Memory:** Proficiency in using pointers for creating and manipulating complex data structures, and managing dynamic memory effectively **Algorithmic Implementation:** Skill in implementing algorithms that interact with data structures, such as sorting, searching, and graph

		algorithms. **Abilities (A)** **Complexity Analysis:** Ability to analyze and understand the complexity of algorithms and data structure operations, and to make trade-offs between different approaches. **Algorithm Design:** Ability to design algorithms that efficiently use data structures to solve complex problems, considering both time and space constraints.
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. **Abilities (A)** **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 highperformance 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	**M23BCE604B**	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**

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: Professional Elective Course (PE)

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 Course (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

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

7. Learning Objectives

S/L	**Learning Objectives**	**Description**

1	**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
M23BCE604B.1	Apply basic concepts of data structures and linear data structure to solve computational problems.
M23BCE604B.2	Apply dynamic memory management techniques using pointers and implement complex data structures.
M23BCE604B.3	Analyze non-linear data structures like trees and graphs.
M23BCE604B.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
M23BCE604B.1	3				3							
M23BCE604B.2	3				3							
M23BCE604B.3		3			3							
M23BCE604B.4		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
Module 2						20
Module 3						20
Module 4						20
Module 5						20
Total	20	20	20	20	20	**100**

Conditions for SEE Paper Setting:

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

10. Future with this Subject

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.

Enhanced Problem-Solving Skills

Competitive Programming: Many programming competitions focus on problems that require a deep understanding of data structures and algorithms. Excelling in these can lead to job offers, internships, and recognition in the tech community.

Technical Interviews: Data structures and algorithms are a central part of technical interviews for software engineering roles. Mastering these in C gives you an edge during the hiring process.

Open Source Contributions: Contributing to open-source projects often requires implementing or improving data structures. This can help you build a portfolio and gain practical experience.

Preparation for Research and Higher Studies

Graduate Studies: If you plan to pursue a Master's or Ph.D., a strong grasp of data structures will be crucial for research in computer science fields such as theory of computation, cryptography, and data mining.

Publications and Innovation: Understanding and potentially innovating in data structures can lead to academic publications, patents, or new technological advancements.

Adaptability to Other Programming Languages

Transition to Other Languages: Once you understand data structures in C, you can easily transition to other languages (like C++, Java, Python) since the concepts remain consistent. This adaptability makes you versatile as a developer.

Understanding Lower-Level Concepts: C is close to the hardware, so learning data structures in C helps you understand how memory is managed and how data structures work at a low level, which is beneficial when working with languages like Rust or Go.

Entrepreneurial Opportunities

Startup Development: If you're interested in developing your software product or startup, efficient data structures will be crucial in ensuring your application can scale and perform well, especially under load.

Product Development: Understanding data structures helps in designing robust, efficient, and scalable products, whether it's an app, a service, or a tool.

6th Semester	Open Elective -I **INTRODUCTION TO DBMS**	M23BCE604C

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 skillto be used forquerying 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 aparticular application. **Attitudes:** To enhance database performance.

3. Syllabus

DataBase Management System SEMESTER – IV			
Course Code	**M23BCE604C**	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**

Course objectives:

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

- To become familiar with database storage structures and access techniques.

Module -1

Introduction to DBMS and Database Design: **8 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: **8hours** 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, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson. *2.* Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw-Hill, 3rdEdition. **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. 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 realworld competencies.
6	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

5. 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.

6. Assessment Details (both CIE and SEE)

Theory Course with 3 Credits: Professional Elective Course (PE)

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 Course (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

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module. • Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

S/L	Syllabus structure	KS Description
1.	Module 1 (Introduction to Data base)	**Knowledge:** Understand the principles of data modeling. **Skills:** Entity-Relationship diagrams (ERDs). These concepts help design efficient and organized database.
2	Module 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 and the foundation of relational databases.
3	Module 3 (SQL)	**Knowledge:** The basics of SQL, the standard language for data query. **Skills:** Writing queries to retrieve, update, and manipulate data.
4	Module 4 (Normalization)	**Knowledge:** Learn about database normalization to eliminate redundancy and improve data integrity. **Skills:** Understand the concept of normalization for optimizing query performance.
5	Module 5 (DataBase applications)	**Knowledge:** Gainsight into query optimization strategies to enhance database performance. **Skills:** To design data base structure for aparticular application.

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

Course Outcomes (COs):

COs	Description
M23BCE604C-1	Understand and apply the basic elements of a relational database management system.
M23BCE604C-2	Apply various constraints, techniques and Structured Query Language (SQL) statement for database operations.
M23BCE604C-3	Analyze various database models and normalization for the given application.
M23BCE604C-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
M23BCE604C.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCE604C.2	3	-	-	-	2	-	-	-	-	-	-	-	3	3
M23BCE604C.3	-	3	-	-	-	-	-	-	-	-	-	-	-	3
M23BCE604C.4	-	-	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						**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:

- 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 INTRODUCTION TO OPERATING SYSTEMS	M23BCE604D

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	Understanding of Operating System Concepts	**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 systemand how they interact, diagnose common OS-related performance issues, and optimize the OS for better resource utilization.
2	Problem-solving Skills	**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	Concurrency and Parallelism	·**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	File Systems	**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.
6	Understanding of Modern Trends	**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.

3. Syllabus

Introduction to Operating systems **SEMESTER – VI**			
Course Code	**M23BCE604D**	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**

Course objectives: This course will enable students to:

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

Module -1

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

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

Module -2

Process management: Process Concept ; Process Scheduling;Operation on Process; Inter-Process Communication.**multithreadedprogramming**:Overview;Multicoreprogramming,multitreadingmodels,threadlibraies,threading issues. **Process scheduling** - Basic Concepts, CPU I/O Burst Cycle; CPU Scheduler – Pre-emptive Scheduling, Dispatcher; Scheduling Criteria; Scheduling Algorithms – FCFS, SJF, RoundRobin, 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, WileyIndia, 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 utilizemultiple 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 concepts.
3	Collaborative Learning	Encourage collaborative learning for improved competency application.
4	Higher Order Thinking (HOTS) Questions:	Pose HOTS questions to stimulate critical thinking related to each competency.
5	Problem-Based Learning (PBL)	Implement PBL to enhance analytical skills and practical application of competencies
6	Multiple Representations	Introduce topics in various representations to reinforce competencies
7	Real-World Application	Discuss practical applications to connect theoretical concepts with real-world competencies.
8	Flipped Class Technique	Utilize a flipped class approach, providing materials before class to facilitate deeper understanding of competencies
9	Programming Assignments	Assign programming tasks to reinforce practical skills associated with competencies.

6. Assessment Details (both CIE and SEE)

Theory Course with 3 Credits: Professional Elective Course (PE)

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 Course (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

- Question paper pattern will be ten questions. Each question is set for 20marks. The medium of the question paper shall be English unless otherwise it is mentioned.
- There shall be 2 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.
- The students have to answer 5 full questions selecting one full question from each module. • Marks scored will be proportionally scaled down to 50 marks

7. Learning Objectives

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

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

Course Outcomes (Cos)

Cos	Description
M23BCE604D.1	Understand the fundamental concepts and principles of operating systems.
M23BCE604D.2	Analyze various inter-process communication, multiprogramming mechanisms and apply different process scheduling algorithms.
M23BCE604D.3	Examine multiple mechanisms for managing deadlock situations and Implement both software and hardware solutions to address the critical-section problem
M23BCE604D.4	Implement and evaluate memory management techniques.
M23BCE604D.5	Examine the structure of file systems and the organization of secondary storage devices

CO-PO-PSO Mapping

Cos/POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO2
M23BCE604D	3	-	-	-		-	-	-	-	-	-	3	3	
M23BCE604D	-	-	3	2	3	-	-	-	-	-	-	3	3	
M23BCE604D	-	-	3	2	3	-	-	-	-	-	-	3		3
M23BCE604D	3	-	3	2	3	-	-	-	-	-	-	3	3	
M23BCE604D	-	-	3	-		-	-	-	-	-	-	-	3	

9. Assessment Plan

Continuous Internal Evaluation (CIE)

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

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

Semester End Examination (SEE)

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

Conditions for SEE Paper Setting:

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

10. Future with this Subject

- Continued 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	M23BCE605

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 Project Work 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
M23BXX605.1	Conduct a comprehensive literature survey and synthesize key findings.
M23BXX605.2	Define a research problem statement based on literature review.
M23BXX605.3	Develop and present a well-structured project report.

CO-PO-PSO Mapping

COs/POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2

M23BXX605.1	3	3		3		3							3	3
M23BXX605.2		3	3		3			3					3	3
M23BXX605.3	3		3	3			3		3	3	3	3	3	3
M23BXX605	**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 Course Laboratory(PCL) MACHINE LEARNING LABORATORY	M23BCEL606

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 scikitlearn 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 biasvariance 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	**M23BCEL606**	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 of Practical Session**	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-textdataset

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.

1. **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/writeup part to be made zero.
4. Duration of SEE shall be 3 hours. **2. 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 problemsolving. 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
M23BCEL606.1	Prepare data for machine learning by effectively applying preprocessing, feature engineering, and exploratory analysis techniques to diverse datasets.
M23BCEL606.2	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.
M23BCEL606.3	Analyze machine learning results by interpreting model outputs, identifying patterns and insights in data, and effectively communicating findings through visualizations and reports.
M23BCEL606.4	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.
M23BCEL606.5	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
M23BCEL606.1	3				3									
M23BCEL606.2		3			3									
M23BCEL606.3				3						3				
M23BCEL606.4				3										
M23BCEL606.5			3											
M23BCEL606	**3**	**3**	**3**	**3**	**3**					**3**				

9. 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, datadriven 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(AE) PROGRESSIVE APP DEVELOPMENT	M23BCE607A

1. Prerequisites

S/L	Proficiency	Prerequisites
1	Basic Web Development (HTML, CSS, JavaScript)	Proficiency in HTML for structuring content, CSS for styling and layout, and JavaScript for adding interactivity to web pages. Familiarity with modern JavaScript (ES6+ features) is expected.
2	Understanding of Web Browsers and Networking	Knowledge of how web browsers render pages, execute JavaScript, and manage client-server communication. Basic understanding of HTTP/HTTPS protocols and RESTful APIs is required.
3	Experience with Version Control (Git)	Competence in using Git for version control, including operations like commit, branch, merge, and resolve conflicts.. Experience with platforms like GitHub or GitLab is advantageous.
4	Familiarity with Asynchronous JavaScript (AJAX, Promises)	Understanding of how to perform asynchronous operations in JavaScript, including making AJAX requests and handling Promises Knowledge of async/await syntax is also beneficial.
5	Basic Knowledge of Web Application Architecture	Understanding of the overall structure and components of web applications, including front-end and back-end interactions, as well as MVC (Model View-Controller) or MVVM (Model-View-ViewModel) patterns.
6	Experience with Responsive Design	Ability to create responsive web designs that work across different devices and screen sizes. Familiarity with CSS frameworks like Bootstrap or Materialize for building responsive layouts is helpful.

2. Competencies

S/L	Competency	KSA Description
1	Understanding PWA Architecture	**Knowledge:** Acquire comprehensive knowledge of the architecture and components of Progressive Web Apps, including service workers, app shells, caching strategies, and offline capabilities. **Skills:** Develop the ability to design and structure Progressive Web Apps effectively, ensuring that all components work together seamlessly to provide a smooth user experience. **Attitudes:** Foster a deep appreciation for the importance of a well-architected PWA and a commitment to adhering to best practices in design and development.

2	Implementing Offline Functionality	**Knowledge:** Understand the principles and technologies that enable offline functionality in PWAs, such as service workers, caching mechanisms, and IndexedDB. **Skills:** Gain the technical expertise to implement and manage service workers, ensuring that web applications can function effectively without an internet connection. **Attitudes:** Develop a proactive approach to creating resilient web applications that provide a consistent user experience, regardless of network conditions.
3	Optimizing PWA Performance	**Knowledge:** Learn the key factors that affect PWA performance, including load times, responsiveness, and efficient resource management. **Skills:**
		Master the use of performance optimization tools like Google Lighthouse and techniques such as lazy loading, code splitting, and efficient caching. **Attitudes:** Cultivate a meticulous attitude towards continuous improvement, always striving to enhance the performance and user experience of web applications.
4	Ensuring Security in PWAs	**Knowledge:** Understand the security challenges specific to web applications and PWAs, including data protection, secure communication, and threat prevention. **Skills:** Develop the skills to implement security best practices, such as HTTPS, secure data handling, and protection against common web vulnerabilities like XSS and CSRF. **Attitudes:** Install a security-first mindset, prioritizing the protection of user data and the integrity of the web application in every aspect of development.
5	Advanced PWA Features and Deployment	**Knowledge:** Gain in-depth knowledge of advanced PWA features, such as push notifications, background sync, and IndexedDB, along with deployment strategies for live environments. **Skills:** Develop the ability to integrate these advanced features into PWAs and manage the deployment process, ensuring that the application is accessible and installable across devices. **Attitudes:** Embrace a forward-thinking approach, eager to explore and implement the latest PWA features and deployment practices to enhance the reach and functionality of web applications.

3. Syllabus

Progressive App Development SEMESTER – VI			
Course Code	**M23BCE607A**	CIE Marks	**50**
Number of Lecture Hours/Week(L: T: P: S)	**(0:0:2)**	SEE Marks	**50**
Total Number of Lecture Hours	**24 hours Practical**	Total Marks	**100**
Credits	**01**	Exam Hours	**03**

Course Objectives:

- Understand and Apply Core Web Technologies.
- Develop and Implement Service Workers
- Optimize Web Application Performance
- Enhance Web Applications with Advanced Features
- Deploy and Secure Progressive Web Apps

Program -1
Create a basic web page with HTML, Style the page using CSS for a responsive layout, Add interactivity using JavaScript.
Program -2
Register a service worker, Create a service worker script that caches static assets, Test the service worker by loading the app offline.
Program -3
Create an app shell structure with HTML and CSS, Implement lazy loading for content within the app shell, Ensure the app shell loads quickly, even on slow networks.
Program -4
Implement a Cache-First strategy for static assets, Implement a Network-First strategy for dynamic content, Test and compare the performance of each strategy.
Program -5
Set up IndexedDB in the web application, Store form data in IndexedDB when offline, Sync the data with a remote server when the network is available.
Program -6
Set up push notifications using the Push API, Customize notification appearance and behaviour, Handle notification clicks and interactions.
Program -7
Audit the web application using Lighthouse, Identify performance bottlenecks and areas for improvement, Implement optimizations. Optimize the performance of a PWA using Lighthouse.
Program -8
Test the application on multiple browsers, Implement feature detection to provide fallbacks for unsupported features; Ensure the application is accessible and inclusive.
Program -9
Set up HTTPS for the local development environment, Secure service workers and sensitive data transmissions,Test the application for common security vulnerabilities.
Program -10
Set up routing for a Single Page Application (SPA), Implement state management using a library, Handle navigation and state changes efficiently.
Program -11
Prepare the PWA for production, Deploy the application on a hosting platform, Make the PWA installable and test the installation process on different devices.
Program -12
Develop the PWA using best practices for service workers, caching, offline functionality, and performance optimization. Test and deploy the PWA, ensuring it meets all PWA criteria.

Developed using common web technologies including HTML, CSS, JavaScript, and WebAssemblywith any platform with a standards-compliant browser, including desktop and mobile devices

TEXTBOOKS:

1. Building Progressive Web Apps *by Tal Ater, O'Reilly Media*

2. Dean Hume, "Progressive Web Apps", Apress 1st Edition (2017)

REFERENCE BOOKS:

1. Learning Progressive Web *Apps* by John M. Wargo, Addison-Wesley Professional
2. JavaScript: The Definitive Guide *by David Flanagan, ,* O'Reilly Media
3. Adaptive Web Design: Crafting Rich Experiences with Progressive *Enhancement* by Aaron Gustafson, New Riders

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	Week 1: Introduction to Progressive Web Apps	Overview of PWAs Key components: service workers, caching, app shell Lab: Set up a basic web application project
2	Week 2: HTML & CSS Fundamentals	Structure and styling of web applications Responsive design principles Lab: Create a responsive layout for a PWA
3	Week 3: JavaScript Basics for PWAs	Core JavaScript concepts and ES6 features Introduction to JavaScript in web development Lab: Implement interactive features using JavaScript
4	Week 4: Introduction to Service Workers	Understanding service workers and their role Basics of service worker registration and installation Lab: Implement a basic service worker to cache assets
5	Week 5: Advanced Service Workers	Handling fetch events and caching strategies Managing updates and background sync Lab: Implement advanced caching strategies and background sync
6	Week 6: Offline Functionality and Data Storage	Using IndexedDB for offline storage Understanding data persistence in PWAs Lab: Implement IndexedDB for storing user data offline
7	Week 7: Push Notifications	Introduction to push notifications Implementing push notifications and managing permissions Lab: Set up push notifications in a PWA
8	Week 8: Performance Optimization	Techniques for optimizing PWA performance Using tools like Google Lighthouse Lab: Analyze and optimize a PWA's performance using Lighthouse
9	Week 9: Security in PWAs	Ensuring security in web applications Implementing HTTPS and secure data handling Lab: Secure a PWA with HTTPS and review security best practices
10	Week 10: Design and UX for PWAs	Best practices for user experience and design Designing a seamless and engaging PWA interface Lab: Design and implement an engaging user interface for a PWA
11	Week 11: Deployment Strategies	Deploying PWAs to live environments Managing updates and version control Lab: Deploy a PWA to a cloud platform and manage updates

12	Week 12: Project Presentation and Review	Final project presentations Peer reviews and feedback Course wrap-up and discussion Lab: Present final PWA projects and receive peer feedback

4. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Assign real-world projects	**Assign real-world projects** where students design and develop Progressive Web Apps.
2	Interactive Workshops	**Conduct workshops** that involve coding sessions and live demonstrations. Focus on specific aspects like service worker setup, caching strategies, or performance optimization, and encourage students to work through exercises in real-time..
3	Case Studies and Examples	**Present case studies** of successful Progressive Web Apps and analyze their features and implementations.
4	Peer Reviews and Group Work	**Facilitate peer review sessions** where students present their projects and provide feedback to each other. Encourage group work on lab assignments to foster collaboration and diverse problem-solving approaches.
5	Tutorials and Guided Labs	**Provide step-by-step tutorials** and guided lab sessions for complex topics like implementing push notifications or optimizing performance.
6	Tool-Based Learning	**Integrate tools and platforms** such as Google Lighthouse for performance auditing and code editors for development.

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/writeup part to be made zero.
4. Duration of SEE shall be 3 hours.

7. Learning Objectives

S/L	Learning Objectives	Description
1	Understand PWA Fundamentals	**Learn** the core concepts and components of Progressive Web Apps, including service workers, caching, and offline functionality.
2	Implement Service Workers	**Develop** the ability to create and configure service workers to manage caching strategies and enable offline access in web applications.
3	Utilize Advanced Features	**Integrate** advanced PWA features such as push notifications, IndexedDB for offline storage, and background synchronization into web applications.
4	Optimize Web Application Performance	**Analyze** and **apply** performance optimization techniques using tools like Google Lighthouse to improve the speed and efficiency of Progressive Web Apps.
5	Ensure Web Application Security	**Implement** security best practices and HTTPS to protect Progressive Web Apps from vulnerabilities and ensure secure data handling.
6	Deploy and Test PWAs	**Design**, **deploy**, and **test** Progressive Web Apps in real-world environments, ensuring they meet performance, accessibility, and usability standards.

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

Course Outcomes (COs)

COs	Description
M23BCE607A.1	Understand and apply the key elements of Progressive Web Apps, such as service workers and caching.
M23BCE607A.2	Analyse the working of service workers and caching to improve offline functionality and performance.
M23BCE607A.3	Implement service workers and caching strategies to build functional Progressive Web Apps.
M23BCE607A.4	Evaluate and optimize PWA performance and security using tools like Google Lighthouse.
M23BCE607A.5	Design and deploy a complete Progressive Web App with advanced features like push notifications and IndexedDB.

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCE607A.1	3	-	-	-	-	-	-	-	-	-	-	-	3	-
M23BCE607A.2	-	3	-	-	-	-	-	-	-	-	-	-	-	3

M23BCE607A.3	-	-	3	-	-	-	-	-	-	-	-	-	3	-
M23BCE607A.4	-	-	-	3	-	-	-	-	-	-	-	-	-	3
M23BCE607A.5	-	-	-	-	3	-	-	-	-	-	-	-	3	3
M23BCE607A	**3**	**3**	3	3	3	-	-	-	-	-	-	-	3	3

9. Assessment Plan

Assessment Plan (For PCL)
Continuous Internal Evaluation (CIE)

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

Semester End Examination (SEE)

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

10. Future with this Subject:

- **Career Opportunities:** With expertise in PWAs, you can work as a front-end or full-stack web developer. Many companies are looking to create PWAs to enhance user experiences, especially on mobile devices.
- **Technical Growth:** Understanding PWAs will naturally lead you to explore other advanced web technologies, such as WebAssembly, Progressive Enhancement, or server-side rendering (SSR). The skills gained from PWA development can be applied to other cross-platform frameworks like React Native, Flutter, or Ionic, allowing you to build applications that work seamlessly across both web and mobile platforms.
- **Industry Trends:** The industry is increasingly adopting PWAs due to their ability to deliver app-like experiences directly through the web. Your skills will be in demand as more businesses look to create or convert their existing websites into PWAs. Staying current with web standards and best practices will be crucial, as PWAs are part of the broader trend towards making the web more powerful, reliable, and engaging.
- **Continuous Learning:** With a solid foundation in PWAs, you might explore advanced frameworks and libraries like React, Angular, or Vue.js, which are often used to build sophisticated PWAs.As you move forward, learning about continuous integration/continuous deployment (CI/CD) practices, cloud services, and containerization (e.g., Docker) could be the next step in your technical growth.
- **Mastering Analytics and Monitoring:** Understanding user behavior through analytics and implementing monitoring tools (e.g., Google Analytics, Firebase) will allow you to continually improve the performance and user engagement of your PWAs.

6th Semester	**Ability Enhancement(AE)** **DEVOPS**	**M23BCE607B**

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**	**Knowledge:** Understand the lifecycle of software builds and

	and Deployment Technologies	deployments, including the use of Maven, Docker, and Jenkins. **Skills:** Ability to apply Maven lifecycle commands, create Docker 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	**M23BCE607B**	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:

Understanding and Application of Version Control Systems.
Mastering Continuous Integration and Automation.
Proficient Use of Automation Tools for Web Application Testing.
Integration and Management of Build and Deployment Pipelines
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 Straub Available 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, cron jobs, 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	Apply Maven Life Cycle Commands on Project and Integrate with Jenkins Tools Introduction to Maven, Maven lifecycle phases, and integrating Maven with
	Maven	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.

2. 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 handson 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 DevOps toolchain 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.

3. 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**

- 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	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 DevOps Toolchain Framework	Students will be able to design, develop, and implement a cohesive DevOps toolchain framework that integrates multiple tools such as Git, Jenkins, Docker, 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
M23BCE607B.1	Apply Version Control Techniques Demonstrate the ability to create, manage, and maintain local and remote repositories using Git and GitBash
M23BCE607B.2	Implement Continuous Integration and Automation Utilize Jenkins to automate the integration, testing, and deployment processes by creating and scheduling CI/CD pipelines.
M23BCE607B.3	Develop Automation Scripts for Web Applications
M23BCE607B.4	Integrate Development Tools with Docker and Maven
M23BCE607B.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
M23BCE607B.1	3	3	-	-	3	-	-	-	-	-	-	-	3	-
M23BCE607B.2	3	3	3	-	3	-	-	-	-	-	-	-	-	3
M23BCE607B.3	3	-	3	-	3	3	-	-	-	-	-	-	3	-
M23BCE607B.4	3	3	3	3	3	-	-	-	-	-	-	-	-	3
M23BCE607B.5	**3**	**3**	3	3	3	3	-	-	3	-	-	-	3	3
M23BCE607B	**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:

- **High Demand for DevOps Professionals:** Proficiency in tools like Git, Jenkins, Docker, and Selenium, coupled with Agile and DevOps methodologies, 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 DevOps Beyond 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.

6th Semester	**Ability Enhancement- V** **PREDICTIVE ANALYTICS USING R**	**M23BCE607C**

1. Prerequisites

S/L	Proficiency	Prerequisites
1	**Programming Fundamentals**	Understand the fundamental syntax of C, including variables, control structures (if, for, while), and basic data structures (arrays, structures). How to read from and write to files, as data input and output are crucial for handling datasets.
2	**Mathematics and Statistics**	Understand measures of central tendency (mean, median) and dispersion (variance, standard deviation). Basic concepts of probability, probability distributions, and statistical inference.
3	**Machine Learning Basics**	Knowledge of evaluation metrics like accuracy, precision, recall, F1-score, and ROC curves.
4	**Data Handling**	Familiarity with more advanced data structures like linked lists and trees can be beneficial. Basic skills in data preprocessing, such as normalization and handling missing values.
5	**Development Environment**	Skills in debugging and profiling tools to analyze the performance of your C code. Hands-on experience with implementing predictive models, even in simpler languages or environments, will help in applying similar techniques in C.

2. Competencies

S/L	Competency	KSA Description

1	**Introduction on Predictive analytics using R:**	**Knowledge:** Understand what predictive analytics is, its goals, and how it is used to forecast future outcomes based on historical data. Understanding of commonly used R packages for data manipulation, visualization and machine learning **Skills:** Techniques for reading data from various sources (CSV, Excel, databases) and writing data to files. Ability to write and debug R code effectively. Familiarity with RStudio for coding and project management. **Attitudes:** Ability to write and debug R code effectively. Familiarity with RStudio for coding and project management.
2	**Basic statistics:**	**Knowledge:** Understanding measures such as mean, median, mode, variance, and standard deviation. Knowledge of R's syntax, including operators, functions, and control structures. Familiarity with core R packages for statistics. **Skills:** Ability to compute and interpret mean, median, variance, and standard deviation using R. Creating visual representations of data using ggplot2 to display distributions, trends, and relationships. **Attitudes:** Ability to approach problems systematically and apply appropriate statistical methods. Ability to explain statistical results and insights clearly to both technical and nontechnical audiences.
3	**Data manipulation:**	**Knowledge:** Knowledge of techniques to handle missing values, outliers, and erroneous data. Concepts related to reshaping and summarizing data, including pivoting, merging, and aggregating.
		Skills: To handle a wide range of data manipulation tasks in R, leading to more effective and efficient data analysis. **Attitudes:** Explore different data manipulation techniques and R packages to find the most effective methods. Ability to approach data manipulation challenges systematically and find efficient solutions.
4	**Predictive Modelling technique:**	**Knowledge:** Understanding the process of building models to make predictions and the importance of data quality and feature selection. Knowledge of various predictive modeling techniques, including regression, classification, and clustering. **Skills:** Ability to preprocess data, handles missing values, and performs feature engineering. Skills in selecting and engineering features that contributes to model accuracy and performance. **Attitudes:** Ability to approach modeling challenges methodically and applies appropriate techniques to address specific problems.

5	**Regression:**	**Knowledge:** Understanding the basic principles of regression analysis, including the purpose of modeling and the interpretation of regression coefficients. Knowledge of how to specify, fit, and validate regression models. **Skills:** Ability to handle missing values, outliers, and data inconsistencies before fitting a regression model. Ability to interpret and explain the coefficients of the regression model in the context of the problem. **Attitudes:** Explore different regression techniques and stay updated with advancements in modeling methods. Ability to approach regression problems methodically and apply appropriate techniques to derive meaningful insights.

3. Syllabus

PREDICTIVE ANALYTICS USING R SEMESTER – VI			
Course Code	**M23BCE607C**	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:

- To learn about R and how Ris used for Predictive modeling.
- To design statistical experiments and analyze the results using modern methods.
- To learn Data manipulation methods and predictive Modeling techniques in R.
- To develop constructive approach to solve business queries with R.

Program -1

Download and install R-Programming environment and install basic packages using install.packages() command in R.

Program -2

Write a program to import the csv file and reading the csv file and extracting some specific information from the data frame.

Program -3

Create a CSV file having Speed and Distance attributes with 50 records. Write R program to draw i) Box plots ii) Histogram iii) Line Graph iv) Multiple line graphs v) Scatter plot to demonstrate the relation between the cars speed and the distance.

Program -4

Write a program to load a dataset into the appropriate data structure and Use built-in functions to compute mean, median, and standard deviation on specific column.

Program -5

Calculate skewness, kurtosis, correlation, and generate visualizations for given dataset details Data1: A sample of 100 measurements with a normal distribution (mean = 50, standard deviation = 10). Data2: A sample of 100 measurements with a normal distribution but different variance (mean = 50, standard deviation = 15). Data3: A sample of 100 measurements that is linearly related to Data1 with some added noise.
Program -6
Write a program to evaluate its performance, and perform hypothesis testing on the model coefficients using linear regression model by using any built in dataset
Program -7
Write a R script to perform both t-test and z-test using any built in dataset and compare the result.
Program -8
R proram to perform linear regression using a household dataset. The dataset will include features like Income, HouseholdSize, Age, and Expenditure, with the goal of predicting Expenditure based on these features.
Program -9
Build and evaluate logistic regression model to create binary classification.
Program -10
Write a R script to perform clustering to determine the number of cluster and visualize the data.
TEXTBOOKS: 1.Learning predictive Analytics with R by Eric Mayor , packt publishing 2. R for Data Science Book by Garrett Grolemund and Hadley Wickham **Reference Books:** 1. Jared P Lander, R for everyone: advanced analytics and graphics, Pearson Education, 2013 2. W. N. Venables, D. M. Smith and the R Core Team, "An Introduction to R", 2013.

4. Syllabus Timeline

S/L	Syllabus Timeline	Description
1	**Week 1: Introduction**	Install & Configure How to load and install packages and other basic operation
2	**Week 2: Read and writing a files**	Techniques for reading data from various sources (CSV, Excel, databases) and writing data to files
3	**Week 3: Data visualization**	Understand and communicate data insights Packages to create various types of visualizations
4	**Week 4-5 Basic statistics**	Understanding measures such as mean, median, mode, variance, and standard deviation and various other statistic in R
5	**Week 6-7 Predictive**	Understanding the process of building models to make predictions and the importance of data quality and feature selection
	Modelling technique	
6	**Week 8-10: Regression**	Understanding the basic principles of regression analysis, including the purpose of modeling and the interpretation of regression coefficients.
7	**Week 11-12: Project based and evaluation**	Final project presentations Peer reviews and feedback Course wrap-up and discussion

5. Teaching-Learning Process Strategies

S/L	TLP Strategies:	Description
1	Assign real-world projects	**Assign real-world projects** where students design and develop Progressive Web Apps.
2	Interactive Workshops	**Conduct workshops** that involve coding sessions and live demonstrations. Focus on specific aspects like service worker setup, caching strategies, or performance optimization, and encourage students to work through exercises in real-time..
3	Case Studies and Examples	**Present case studies** of successful Progressive Web Apps and analyze their features and implementations.
4	Peer Reviews and Group Work	**Facilitate peer review sessions** where students present their projects and provide feedback to each other. Encourage group work on lab assignments to foster collaboration and diverse problem-solving approaches.
5	Tutorials and Guided Labs	**Provide step-by-step tutorials** and guided lab sessions for complex topics like implementing push notifications or optimizing performance.
6	Tool-Based Learning	**Integrate tools and platforms** such as Google Lighthouse for performance auditing and code editors for development.

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**

- 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.

4. Learning Objectives

S/L	Learning Objectives	Description
1	Introduction on Predictive analytics using R:	Build a strong foundation in R programming, enabling you to tackle a variety of data analysis tasks and apply your skills in diverse domains.
2	Basic statistics	Enabling to analyze and interpret data effectively and make informed decisions based on statistical evidence.
3	Data manipulation	Skills to effectively handle, transform, and clean data. This is crucial for preparing data for analysis and ensuring accurate results.
4	Predictive Modelling technique	Understanding and applying methods to make predictions based on data. These objectives encompass both the theoretical aspects of predictive modelling and practical skills for implementing models using various techniques.
5	Regression	Understanding of regression techniques and their applications, enabling you to perform effective and accurate predictive modelling.

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

Course Outcomes (COs)

Cos	Description
M23BCE607C.1	Understand and apply fundamental concepts like variables, datatypes and commands
M23BCE607C.2	Apply statistical and predictive analysis methods to real life scenario
M23BCE607C.3	Analyze the performance of models using appropriate evaluation
M23BCE607C.4	Evaluate the appropriateness and validity of models and report the results
M23BCE607C.5	Design various experiment based on graph and charts for data visualization

CO-PO-PSO Mapping

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
M23BCE607C.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE607C.2	3	-	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE607C.3	-	3	-	-	-	-	-	-	-	-	-	-	3	3
M23BCE607C.4	-	-	3	-	-	-	-	-	-	-	-	-	3	3
M23BCE607C.5	-	-	-	3	3	-	-	-	-	-	-	-	3	3
M23BCE607C	3	3	3	3	3	-	-	-	-	-	-	-	3	3

6. 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**

7. Future with this Subject:

1.**Integration with Emerging Technologies**

- Machine Learning and AI: Continued integration of R with machine learning and AI frameworks, including advanced techniques such as deep learning and reinforcement learning. R's ecosystem will increasingly incorporate libraries and tools for these advanced techniques.

2. Advanced Data Visualization

- Interactive Visualizations: Growing use of interactive and dynamic data visualization tools such as plotly, shiny, and ggiraph to provide more engaging and user-friendly data exploration experiences.
- Augmented Analytics: Integration of augmented analytics tools that leverage AI to automate and enhance data visualization and insights.

3. Automated Machine Learning (AutoML)

- Simplified Modeling: Development of AutoML tools and packages in R to simplify the process of building, tuning, and deploying machine learning models. This will make predictive analytics more accessible to nonexperts.
- Model Selection and Tuning: Enhanced AutoML capabilities to automatically select and tune the best models based on the given data and problem. **4. Enhanced Data Handling and Processing**
- Big Data Integration: Improved capabilities for handling and analyzing big data through integration with big data technologies like Hadoop and Spark, using packages such as sparklyr.
- Real-Time Analytics: Increased focus on real-time data processing and streaming analytics to support timesensitive decision-making.

5. Expansion of Predictive Analytics Applications

- Industry-Specific Solutions: Growth in specialized predictive analytics applications tailored to specific industries such as healthcare, finance, retail, and manufacturing, with R packages and frameworks designed for these domains.
- Predictive Maintenance: Enhanced tools for predictive maintenance and failure prediction in industrial settings, leveraging sensor data and IoT technologies.

6. Improved Model Interpretability

- Explainable AI: Development of better tools and methods for explaining and interpreting complex predictive models, including SHAP, LIME, and other model-agnostic techniques.

6thSemester	Ability Enhancement Course(AEC) GOLANG PROGRAMMING	M23BCE607D

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	**M23BCE607D**	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 learn the basics of Golang Programming .
- To understand and develop Go applications using syntax, data structures, and standard libraries.
- To Debug and analyze Go code for logic, performance, and concurrency issues.
- 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**

- 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	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.

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

Course Outcomes (COs)

COs	Description
M23BCE607D	Apply the basic programming Go Lang constructs to develop standalone applications.
M23BCE607D	Apply the concept of functions and recursive functions in GoLang programming
M23BCE607D	Develop applications using Go Routines and channels
M23BCE607D	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
M23BCE607D	**3**												**2**	
M23BCE607D		**3**											**2**	
M23BCE607D			**3**											**2**
M23BCE607D				**3**	**3**									**2**

9. Assessment Plan

Continuous Internal Evaluation (CIE)

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

Semester End Examination (SEE)

	CO1	CO2	CO3	CO4	CO5	**Total**

Module 1						
Module 2						
Module 3						
Module 4						
Module 5						
Total						**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 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:

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, highperformance 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.

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.

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.

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.

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 resourceconstrained environments like IoT devices.

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.

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.

6thSemester	Non-Credit Mandatory Course(NCMC) **PHYSICAL EDUCATION (SPORTS & ATHLETICS)**	M23BPEK608

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
3rd Sem 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.
4th Sem 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:

- 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 socialharmony in general.

Pedagogy–Guidelines

Sl No	Topic	roup size	Location	Activity execution	Reporting	valuation 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 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

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/ 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
5.	Preparing an actionable business proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/ City Areas / Grama panchayat/ public associations/ Government Schemes officers /campus etc…	Group selection/ proper consultation / Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
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	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemes officers	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
	approaches.		/campus etc…			

8.	Contribution to any nationallevel 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 Schemesoffice rs /campus etc…	Group selection/ proper consultation/ Continuous monitoring / Information board	Report should be submitted by an individual to the concerned evaluation authority	Evaluation as per the rubrics of the scheme and syllabus by NSS officer
9.	Spreading public awareness under rural out reach programs.(mini mum 5 programs).Soci al connect and responsibilities.	May be individual or team	Villages/City Areas / Grama panchayat/ public associations/ Government Schemesoffice rs /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 Schemesoffice rs /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.(Min imum 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**	
CaseStudy-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.

6thSemester	Non-Credit Mandatory Course(NCMC) PHYSICAL EDUCATION (SPORTS & ATHLETICS)	M23BPEK608

Non-Credit Mandatory Courses(NCMC) PHYSICAL EDUCATION (SPORTS & ATHLETICS)			
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	**-**

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

CO1.Understand the fundamental concepts and skills of Physical Education, Health, Nutrition and Fitness.

CO2.Familiarization of health-related Exercises, Sports for overall growth and development.

CO3.Create a foundation for the professionals in Physical Education and Sports

. CO4.Participate in the competition at regional/state/national/international levels.

CO5.Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.

Semester-VI

PHYSICAL EDUCATION (SPORTS & ATHLETICS)—II

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

CO1. Understand the ethics and moral values in sports and athletics

CO2.Perform in the selected sports or athletics of the student's choice.

CO3.Understand the roles and responsibilities of organization and administration of sports and games.

Module-1

Ethics and Moral Values

A. Ethics in Sports

B. Moral Values in Sports and Games

Module-2

Specific Games(Anyone to be selected by the student)

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

Sl. No.	Activity	Marks
1.	Participation of student in all the modules	20

2.	Quizzes—2,each of 15 marks	**30**
3.	Final presentation/exhibition/Participation in competitions/practical on specific tasks assigned to the students	**50**
	Total	**100**

6thSemester	**Non-Credit Mandatory Course(NCMC)** **YOGA**	**M23BYOK608**

Non-Credit Mandatory Courses(NCMC) **Yoga**			
Course Code	**M23BYOK608**		
NumberofLectureHours/Week(L:T:P:S)	**0:0:2:0**	CIEMarks	**100**
TotalNumberof LectureHours	-	SEEMarks	**-**
Credits	**0**	TotalMarks	**100**
EvaluationMethod:ObjectivetypeTheory/Pract ical/Viva-Voce			

Courseobjectives:

1. Toenable the student to have good Health.
2. To practice mental hygiene.
3. To possesse motionalst ability.
4. To integrate moral values.
5. To attain a higher level of consciousness.

TheHealthBenefitsofYoga

The benefits of various yoga techniques have been supposed to improve

- Body flexibility,
- performance,
- stressreduction,
- attainment of inner peace,and
- Self-realization.

The system has been advocated as a complementary treatment to aid the healing of several ailments suchas

- coronaryheart disease,
- depression,
- anxietydisorders,
- asthma,and
- extensive re habilitation for disorders including musculo skeletal problems and traumatic brain injury. The systemhas also been suggested as behavioral therapy for smoking cessation and substance abuse(includingalcoholabuse).

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

- Physical
  1. Improved body flexibility and balance
  2. Improved cardio vascular endurance(strongerheart)
  3. Improved digestion
  4. Improved abdominal strength
  5. Enhanced overall muscular strength
  6. Relaxation of muscularstrains
  7. Weigh tcontrol
  8. Increased energy levels
  9. Enhanced immunesystem
- 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

Semester VI

- AshtangaYoga
  1. Dharana
  2. Dhyana(Meditation)
  3. Samadhi
- Asana by name,technique,precautionary measures and benefits of each asana
- Different types of Asanas

  a. Sitting1.Bakasana2.Hanumanasana 3.EkapadaRajakapotasana4. YogamudrainVajrasana b. Standing1. Vatayanasana2. Garudasana

  c. Balancing1.Veerabhadrasana2. Sheershasana

  d. Supineline1.Sarvangasana2.SetubandhaSarvangasana3.Shavasanaa (Relaxation poisture).
- Revision of Kapalabhati practice 80strokes/min-3rounds
- Different types.Meaning by name, technique, precaution ary measures and benefits of each Pranayama1.Bhastrika2. Bhramari
- Meaning, Need, importanceofShatkriya.

- Different types.Meaning by name, technique, precautionary measures and benefits of each Kriya1.Jalaneti&sutraneti2.Nouli(onlyfor men)3. SheetkarmaKapalabhati

Course outcomes(CourseSkillSet):
At the end of the course,thestudentwill beableto:

- Understand the meaning, aim and objectives of Yoga.
- Perform Surya namaskar and able toTeach its benefits.
- Understand and teach different Asanas by name,it simportance,methds and benefits.
- Instruct Kapala bhati and it sneed and importance.
- Teach different types of Pranayama by its name,precautions,procedure and uses
- Coach different types of Kriyas, method follow and usefulness.

AssessmentDetails(bothCIEandSEE)

- Students will be assessed with internal test by a. Multiple choice questions b. Descriptive type questions (Twointernal assessmenttests with 25 marks/test)
- Finaltestshallbeconductedforwholesyllabusfor50marks.
- ContinuousInternal Evaluationshallbefor100marks(includingIAtest)

SuggestedLearningResources:
Books:

- Yoga pravesha in Kannada by Ajitkumar
- Light on Yoga by BKS Iyengar
- Teaching Methods for Yogic practices byDr.MLGharote&Dr.SK Ganguly
- Yoga Instructor Course handbook published by SVYASA University,Bengaluru
- Yoga for Children–step by step –by Yamini Muthanna

Web links and Video Lectures(eResources):links https://youtu.be/KB-TYlgd1wE https://youtu.be/aa-TG0Wg1Ls