



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin-531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute Vision:

To develop highly skilled professionals with strong ethical and human values.

Institute Mission:

M1: To impart quality education through effective teaching–learning practices, industrial exposure, and professional training.

M2: To produce competent and highly knowledgeable graduates with a positive attitude and problem-solving ability.

M3: To instill self-confidence, leadership qualities, and social responsibility to face professional and societal challenges.

Department Vision:

To emerge as a center of excellence in Computer Science and Engineering education and research, producing competent, industry-ready professionals with innovation, ethical values, and social responsibility.

Department Mission:

M1: To impart strong foundational knowledge in computer science, programming, and emerging technologies through effective teaching–learning practices.

M2: To develop problem-solving, analytical, and research skills by encouraging innovation, projects, and industry interaction.

M3: To produce industry-ready professionals with proficiency in modern tools, teamwork, and professional communication.

M4: To inculcate ethical values, social responsibility, and lifelong learning skills to meet professional and societal challenges.

Program Educational Objectives (PEOs):

PEO1: Graduates will apply strong foundations in computer science, mathematics, and engineering principles to analyze and solve real-world problems.

PEO2: Graduates will design, develop, and deploy software systems and applications using modern Engineering tools, technologies, and industry-standard practices.

PEO3: Graduates will pursue successful careers in industry, entrepreneurship, higher education, or research through continuous learning and skill enhancement.

PEO4: Graduates will demonstrate professional ethics, effective communication, teamwork, and leadership while contributing responsibly to society.



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

Program Specific Outcomes (PSOs):

PSO1: Apply core computing concepts to design efficient software and systems.

PSO2: Use modern tools and emerging technologies to solve real-world problems ethically.

Program Outcomes (POs):

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8).



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program: B. Tech - Computer Science and Engineering

Regulation: R24

Course Structure: III Year I Semester (or) Semester-V

S. No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	PC	R24CSPC15	Computer Networks	3	0	0	3
2	PC	R24CSPC16	Compiler Design	3	0	0	3
3	PC	R24CSPC17	Competitive Programming	3	0	0	3
4	PE-I	R24CSPE01 R24CSPE02 R24CSPE03 R24CSPE04 R24CSPE05	1. Information Retrieval and Search Engines 2. Data Warehousing and Data Mining 3. Human Computer Interaction 4. No SQL Databases 5. 12-Week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
5	OE-I	R24CSOE01 R24CSOE02 R24CSOE03 R24CSOE04 R24CSOE05	1. Data Structures 2. Database Management Systems 3. Agile Methodologies 4. Operating Systems 5. Scripting Languages	3	0	0	3
6	PC	R24CSPC18	Computer Networks Lab	0	0	3	1.5
7	PC	R24CSPC19	Competitive Programming Lab	0	0	3	1.5
8	SC	R24CSSC03	Full Stack Development-2	0	0	3	1.5
9	PC	R24CSPC20	Advanced Coding Lab-II	0	0	2	1
10	HS	R24HS07	Quantitative Aptitude and Effectual Communication Skills	0	0	2	1
11	CSP	R24CSP01	Evaluation of Community Service Project	0	0	0	1
Total				15	00	13	22.5

Category	Courses	Credits
PC- Professional Core Courses	6	13
PE- Professional Elective Courses	1	03
OE- Open Elective Courses	1	03
HS- Humanity Science and Management Courses	1	01
SC- Skill Oriented Courses	1	1.5
CSP- Community Service Project	1	01
Total	10	22.5



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program: B. Tech Computer Science and Engineering

Regulation: R24

Course Structure: III Year II Semester (or) Semester-VI

S.No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	PC	R24CSPC21	Artificial Intelligence and Machine Learning	3	0	0	3
2	PC	R24CSPC22	Cryptography & Network Security	3	0	0	3
3	HS	R24HS08	Managerial Economic and Financial Accounting	2	0	0	2
4	PE-II	R24CSPE06 R24CSPE07 R24CSPE08 R24CSPE09 R24CSPE10	1. Cloud Computing 2. Augmented Reality and Virtual Reality 3. Cyber Security 4. DevOps 5. 12-Week MOOC Swayam / NPTEL course recommended by the BoS	3	0	0	3
5	OE-II	R24CSOE06 R24CSOE07 R24CSOE08 R24CSOE09 R24CSOE10	1. Computer Networks 2. Mobile Computing 3. Software Project Management 4. Object Oriented Through JAVA 5. Fundamentals of Cloud Computing	3	0	0	3
6	PC	R24CSPC23	Artificial Intelligence and Machine Learning Lab	0	0	3	1.5
7	PC	R24CSPC24	Cryptography & Network Security Lab	0	0	3	1.5
8	SC	R24CSSC04	Prompt Engineering / SWAYAM Plus - Certificate program in Prompt Engineering and ChatGPT	0	1	2	1.5
9	HS	R24HS09	High Level Reasoning and Employability Skills	0	0	2	1
10	ES	R24ES13	Design thinking and Innovation Lab	0	0	2	1
11	AC-I	R24AC01	Technical Paper Writing and Intellectual Property Rights	2	0	0	0
Total				16	01	10	20.5
Mandatory Internship of 08 weeks duration during summer vacation							

Category	Courses	Credits
ES- Engineering Science Courses	1	01
PC- Professional Core Courses	4	09
PE- Professional Elective Courses	1	03
OE- Open Elective Courses	1	03
HS- Humanity Science and Management Courses	2	03
SC- Skill Oriented Courses	1	1.5
Total	11	20.5

R24CSPC15**COMPUTER NETWORKS**
(Computer Science and Engineering)**3 0 0 3****Course Objectives:**

1. To understand the various functionalities of computer networks topologies, architectures and transmission mediums
2. To understand state-of-the-art in network protocols, architectures, and applications.
3. To explore the various layers of OSI Model and TCP / IP Model

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPC1 5.1	Demonstrate knowledge of network models such as OSI, TCP/IP, along with communication techniques, transmission media	3	1	-	-	1	-	-	-	-	-	2	3	1	L1, L2	
R24CSPC1 5.2	Analyze data link layer services, functions and evaluate medium access control protocols including ALOHA, CSMA/CD, CSMA/CA, Token Passing, FDMA, TDMA, and CDMA.	3	3	1	2	2	-	-	-	-	-	2	3	2	L2, L3	
R24CSPC1 5.3	Analyze network layer services, routing algorithms (shortest path, distance vector, link state) and congestion control mechanisms, and apply them to improve network efficiency. <i>(Analyze)</i>	3	2	2	2	2	-	-	-	-	-	2	3	2	L3, L4	
R24CSPC1 5.4	Analyze transport layer protocols, including TCP and UDP, their connection management, flow control, error control, and congestion handling mechanisms, and understand end-to-end communication concepts. <i>(Analyze)</i>	3	2	1	2	2	-	-	-	-	-	2	3	2	L4, L5	
R24CSPC1 5.5	Analyze the working of application layer protocols such as HTTP and DNS, including architecture, message formats, and client-server interactions. <i>(Analyze)</i>	3	2	3	1	1	2	-	-	-	-	1	3	3	L5, L6	

SYLLABUS**UNIT I: Introduction & Physical Layer****15 Hours**

Introduction: Network Topologies and Types: LAN, MAN, WAN. Network Reference models: The OSI Reference Model, The TCP/IP Reference Model. OSI Vs TCP/IP models comparison, Internet History – Arpanet.

Physical Layer: Introduction to Guided Transmission Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and unguided media: Wireless: Radio waves, microwaves, infrared.

Self-Learning Topics: Digital Modulation and Multiplexing. **CO's–CO1**

UNIT II: Data Link Layer and Media Access Control **15 Hours**

Design Issues: Services Provided to the Network Layer

Error Detection and Correction: Error Detection: CRC, Checksum. Error Correction:

Framing: Bit and Byte stuffing; fixed size framing, variable size framing.

Flow Control: Sliding Window Protocols. Stop and wait ARQ; Go Back N, Selective repeat

Channel allocation methods: Channelization: TDM, FDM.

Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance – IEEE standard 802 for LANS –Ethernet,

Controlled Access: Token Bus, Tokenring. Bridges and virtual LANs. **CO's–CO2**

Self-Learning Topics: IEEE 802.11 and 802.16. Data link layer switching,

UNIT III: Network Layer and Congestion Control **15 Hours**

Design Issues : Services Provided to the Transport layer , Store and Forward Packet Switching, Implementation of Connectionless Service-Implementation of Connection Oriented Service-Comparison of Virtual Circuit and Datagram Networks,

RoutingAlgorithms:, The Optimality principle . Shortest path, Flooding, Flow based Distance vector, Link state, Hierarchical, Broadcast routing,

Congestion Control algorithms–General principles of congestion control, Congestion prevention polices, Choke packets, Load shedding, and Jitter Control. **CO's–CO3**

Self-Learning Topics: Internet Working: Tunnelling, internetworking, Fragmentation, Network layer in the internet – IP protocols-IP Version 4 protocol-IPV4 Header Format, IP addresses, Class full Addressing, CIDR, NAT-, Subnets,

UNIT IV: Transport Layer **13 Hours**

The Transport Layer: Elements of transport protocols – addressing, establishing a connection, releasing connection, flow control and buffering and crash recovery, End to end protocols: UDP, Real Time Transport Protocol.

The Internet Transport Protocol: TCP- reliable Byte Stream (TCP) end to end format, segment format, connection establishment and termination, sliding window revisited, adaptive retransmission, TCP extension, Remote Procedure Call. **COs–CO4**

Self-Learning Topics: Internet control protocols: OSPF, BGP, Internet multicasting, Mobile IP, IPV6.ICMP-ARP- DHCP

UNIT V: Application Layer **12 Hours**

WWW and HTTP: Architecture- Client (Browser), Server, Uniform Resource Locator

HTTP: HTTP Transaction, HTTP Operational Model and Client / Server Communication, HTTP Generic Message Format, HTTP Request Message Format, HTTP Response Message Format.

The Domain Name System: The DNSN namespace, Resource Records, Name Servers,

CO's–CO5

Self-Learning Topics: Electronic Mail: Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. **Computer Networks in Cloud Computing** (Cloud networking in platforms like AWS, Azure, GCP; Virtual networks, load balancing, network security, SDN concepts) CO3–CO5, PO1, PO2, PO3, PO5, PO11, PSO1, PSO2
2. **Network Security and Cybersecurity Practices** (Firewalls, intrusion detection systems, secure protocols, network attacks and prevention techniques) CO2, CO3, CO5, PO1, PO2, PO3, PO5, PO11, PSO1, PSO2
3. **Emerging Networking Technologies** (IoT networking, 5G communication, network virtualization, software-defined networking (SDN)) CO1, CO 3, CO5, PO1, PO2, PO3, PO5, PO11, PSO1, PSO2

Text Books:

1. **Dr. Sanjay Sharma**, *Computer Networks*, 2nd Edition, Katson Books, New Delhi, 2019
2. **B. A. Forouzan**, *Data Communications and Networking*, 5th Edition, McGraw Hill Education (India), 2017.
3. *Computer Networks* – Andrew S Tanenbaum, Fifth Edition. Pearson Education / PHI
4. *Computer Networks*, 5ed, David Patterson, Elsevier

Reference Books:

1. *Data Communications and Networks* – Achut S Godbole, Atul Kahate
2. *Computer Networks*, Mayank Dave, CENGAGE
3. **N. N. Sakhare and A. V. Dhumane**, *Computer Network*, 1st Edition, Technical Publications, Pune, 2018.
4. **D. S. Gaikwad and N. D. Lambe**, *Computer Network Technology*, 1st Edition, Technical Publications, Pune, 2016.
5. **Achyut S. Godbole**, *Data Communications and Networks*, 2nd Edition, McGraw Hill Education (India), 2013.
6. *An Engineering Approach to Computer Networks*- S.Keshav, 2nd Edition, Pearson Education

Web References:

1. NPTEL (SWAYAM): <https://nptel.ac.in>
2. GeeksforGeeks: <https://www.geeksforgeeks.org>
3. Cisco Networking Academy: <https://www.netacad.com>
4. Tutorials Point: <https://www.tutorialspoint.com>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is LAN, MAN, and WAN? (Unit I)
2. Define OSI reference model. (Unit I)
3. What is TCP/IP model? (Unit I)
4. List different transmission media. (Unit I)
5. Define CRC and checksum. (Unit II)
6. What is framing in data link layer? (Unit II)
7. Define ALOHA and CSMA. (Unit II)
8. What is routing? (Unit III)
9. Define congestion control. (Unit III)
10. What is TCP? (Unit IV)
11. What is UDP? (Unit IV)
12. Define HTTP and DNS. (Unit V)

L2: Understand

1. Explain OSI vs TCP/IP models. (Unit I)
2. Describe different transmission media. (Unit I)
3. Explain framing techniques (bit stuffing, byte stuffing). (Unit II)
4. Describe error detection techniques (CRC, checksum). (Unit II)
5. Explain sliding window protocols. (Unit II)
6. Describe shortest path and distance vector routing. (Unit III)
7. Explain congestion control mechanisms. (Unit III)
8. Describe TCP connection establishment and termination. (Unit IV)
9. Explain flow control in transport layer. (Unit IV)
10. Explain HTTP request and response format. (Unit V)
11. Describe DNS working mechanism. (Unit V)

L3: Apply

1. Apply OSI model to analyse a networking scenario. (Unit I)
2. Demonstrate how CRC is used for error detection. (Unit II)
3. Apply sliding window protocol for data transmission. (Unit II)
4. Solve a routing problem using shortest path algorithm. (Unit III)
5. Apply distance vector routing algorithm to a network. (Unit III)
6. Demonstrate congestion control using leaky bucket algorithm. (Unit III)
7. Apply TCP sliding window mechanism for data transfer. (Unit IV)
8. Demonstrate TCP connection setup using three-way handshake. (Unit IV)
9. Apply HTTP protocol in client-server communication. (Unit V)
10. Demonstrate DNS resolution process. (Unit V)

L4: Analysing

1. Analyze differences between OSI and TCP/IP models. (Unit I)
2. Compare guided and unguided transmission media. (Unit I)

3. Analyze different MAC protocols (CSMA/CD vs CSMA/CA). *(Unit II)*
4. Compare Stop-and-Wait, Go-Back-N, and Selective Repeat. *(Unit II)*
5. Analyze performance of routing algorithms (link state vs distance vector). *(Unit III)*
6. Examine congestion control techniques and their effectiveness. *(Unit III)*
7. Analyze TCP vs UDP performance differences. *(Unit IV)*
8. Compare different retransmission techniques in TCP. *(Unit IV)*
9. Analyze HTTP protocol behaviour in web communication. *(Unit V)*
10. Compare DNS with other naming services. *(Unit V)*

L5: Evaluating

1. Evaluate the efficiency of OSI model in modern networks. *(Unit I)*
2. Assess advantages and disadvantages of different transmission media. *(Unit I)*
3. Evaluate error detection vs error correction techniques. *(Unit II)*
4. Critically evaluate different MAC protocols. *(Unit II)*
5. Evaluate effectiveness of congestion control algorithms. *(Unit III)*
6. Assess routing algorithms for large-scale networks. *(Unit III)*
7. Evaluate TCP congestion control mechanisms. *(Unit IV)*
8. Assess performance of UDP in real-time applications. *(Unit IV)*
9. Evaluate HTTP protocol efficiency in web applications. *(Unit V)*
10. Assess DNS scalability and reliability. *(Unit V)*

L6: Create

1. Design a network using OSI model concepts. *(Unit I)*
2. Develop a framing technique for efficient data transmission. *(Unit II)*
3. Design a MAC protocol for a given network scenario. *(Unit II)*
4. Develop a routing algorithm for optimized path selection. *(Unit III)*
5. Design a congestion control mechanism for high-traffic networks. *(Unit III)*
6. Create a TCP-based communication model for reliable transfer. *(Unit IV)*
7. Design an efficient transport layer protocol for real-time systems. *(Unit IV)*
8. Develop a client-server application using HTTP protocol. *(Unit V)*
9. Design a DNS system for a distributed network. *(Unit V)*
10. Create a secure application layer protocol for data exchange. *(Unit V)*

Chairperson
Board of Studies (CSE)

R24CSPC16**COMPILER DESIGN****3 0 0 3**

(Common to CSE, CSE (DS), CSE (AI & ML))

Course Objectives:

1. Understand the phases of a compiler and language processing systems.
2. Apply lexical and syntax analysis techniques.
3. Analyze syntax-directed translation and semantic processing.
4. Design intermediate representations and runtime environments.
5. Implement code optimization and code generation techniques.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPC 16.1	Understand the basics of language processing and implement lexical analyzer for any language	3	–	2	–	2	–	–	–	–	1	–	3	–	L1, L2
R24CSPC 16.2	Describe the different types of parsing and implement parser for any language	3	3	2	1	2	–	–	–	1	1	–	3	2	L2, L3
R24CSPC 16.3	Explain the different intermediate code representations and use Syntax directed	3	3	2	2	2	–	–	–	–	1	–	3	2	L2, L4
R24CSPC 16.4	Apply the basics of data flow analysis, optimizations, and run time environment required for handling recursive procedures	3	2	3	2	3	–	–	–	–	1	1	3	2	L3, L4
R24CSPC 16.5	Apply machine-independent and machine-dependent code optimization techniques, data-flow analysis, and target code generation methods to improve the efficiency of compiled programs.	3	3	3	1	2	-	-	-	-	-	-	3	2	L5 L6

SYLLABUS**UNIT-I:****15 Hours**

Introduction To Compiling: The evolution of programming languages and basic language processing system; Phases of a Compiler, Bootstrapping and Simple one pass compiler, Cross Compilers.

Lexical Analysis: The Role of Lexical Analyzer, Input Buffering, and Specification of Tokens: Regular Expressions, Regular Definitions- Recognition of Tokens, A language for specifying Lexical Analyzers (LEX).

COs-C01

Self-Learning Topics: LEX/FLEX tool basics, Regular expressions practice, Token design for

simple languages

UNIT-II:

12 Hours

Syntax Analysis: The role of the Parser, Context free Grammars, Elimination of Left Recursion, Left factoring a grammar

Top-Down Parsing: Recursive descent Parsing, First and Follow, Predictive Parsing, LL (1) Grammars.

Bottom-Up Parsing: Shift-Reduce Parser, LR Parsers SLR, Canonical LR, LALR, Operator Precedence Parser

COs-C02

Self-Learning Topics: YACC/Bison basics, Ambiguous grammars, Parser error handling

UNIT-III:

14 Hours

Syntax-Directed Translation and Semantic Analysis: Syntax directed definition (SDD) Syntax directed translation (SDT) and translation schemes (TS), Synthesized and Inherited Attributes, S-Attributed SDD, L-Attributed SDD

Type Checking: Type Systems, Specification of a Simple type checker, Equivalence of Type Expressions, Type Conversions

COs-C03

Self-Learning Topics: Attribute grammars, Type checking concepts, Syntax tree applications

UNIT-IV: Run-Time Environments and Intermediate Code Generation:

15 Hours

Run-Time Environments: Storage Allocation Strategies: Static Allocation, Stack Allocation, Heap Allocation, Activation records, Access Links, Symbol Tables. Heap management, Introduction to garbage collection

Intermediate Code Generation: Interdiction to intermediate address code: forms of intermediate code, Three Address Code, Implementations, basic blocks, DAG, DAG representation of basic blocks, Applications of DAG, DAG based local optimization.

COs-C04

Self-Learning Topics: Garbage collection, Symbol table implementation

UNIT-V: Code optimization and Target Code Generation Code Generation:

14 Hours

Code optimization: Introduction, Principal sources of optimization; Machine-Independent optimizations: The principal sources of optimization, Basic blocks and flow graphs, Introduction to data-flow analysis, Foundations of data-flow analysis,

Target Code Generation Code Generation: Issues in the design of a Code Generator, the target language, addresses in the target code, Machine-dependent Optimizations: Peephole optimization, Register allocation and assignment, Dynamic Programming code generation.

COs-C05

Self-Learning Topics: Code optimization techniques, Register allocation

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson.
2. Compiler Construction-Principles and Practice, Kenneth C Loudon, Cengage Learning

Reference Books:

1. Modern Compiler Implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
2. The Theory and Practice of Compiler writing, J. P. Tremblay and P. G. Sorenson, TMH
3. lex & yacc, 2nd Edition by John Levine, Doug Brown, Tony Mason

Web Resources:

1. <https://www.edx.org/course/compilers>
2. <https://nptel.ac.in/courses/106/108/106108113/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment–1 (%)	Internal Assessment–2 (%)
L1 (Remember)	35	–
L2 (Understand)	40	–
L3 (Apply)	25	25
L4 (Analyze)	–	35
L5 (Evaluate)	–	25
L6 (Create)	–	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define compiler.
2. What is a token?
3. Define lexical analysis.
4. What is a context-free grammar?
5. Define parsing.
6. What is a syntax tree?
7. What is three-address code?
8. Define activation record.
9. What is a basic block?
10. Define code optimization.

L2: Understand

1. Explain the phases of a compiler.
2. Describe the role of lexical analyzer.
3. Explain FIRST and FOLLOW sets.
4. Describe LL(1) parsing.
5. Explain shift-reduce parsing.
6. Discuss syntax-directed translation.
7. Explain type checking.
8. Describe runtime environment.
9. Explain intermediate code generation.
10. Summarize code optimization techniques.

L3: Apply

1. Construct tokens using regular expressions.
2. Design a lexical analyzer for a simple language.
3. Compute FIRST and FOLLOW for a grammar.
4. Construct predictive parsing table.
5. Apply shift-reduce parsing for a given string.
6. Generate three-address code for expressions.

7. Implement syntax-directed translation scheme.
8. Construct syntax tree for given expression.
9. Apply type checking rules.
10. Perform basic code optimization.

L4: Analysing

1. Compare top-down and bottom-up parsing.
2. Analyze ambiguity in grammars.
3. Compare LL(1) and LR parsing techniques.
4. Analyze syntax tree vs DAG.
5. Compare different intermediate representations.
6. Analyze runtime storage allocation methods.
7. Examine data-flow analysis.
8. Compare optimization techniques.
9. Analyze symbol table implementation.
10. Evaluate parsing conflicts.

L5: Evaluating

1. Evaluate efficiency of parsing techniques.
2. Assess code optimization methods.
3. Evaluate intermediate code representations.
4. Compare compiler design approaches.
5. Assess runtime environment strategies.
6. Evaluate LR parsing methods (SLR vs LALR).
7. Judge effectiveness of peephole optimization.
8. Evaluate symbol table structures.
9. Assess data-flow analysis techniques.
10. Evaluate code generation strategies.

L6: Create

1. Design a lexical analyzer for a given language.
2. Develop a parser for a grammar.
3. Create syntax-directed translation scheme.
4. Design intermediate code generator.
5. Develop runtime environment model.
6. Create optimization algorithm.
7. Design code generator for a target machine.
8. Develop compiler for simple language.
9. Create DAG representation for expressions.
10. Design complete compilation process for mini-language.

Chairperson
Board of Studies (CSE)

Course Objectives:

1. Develop strong problem-solving and logical thinking skills for algorithmic challenges.
2. Understand and apply fundamental data structures and algorithms efficiently.
3. Design optimized solutions within time and space constraints.
4. Improve coding speed, accuracy, and debugging skills under time pressure.
5. Apply techniques such as greedy, divide and conquer, dynamic programming, and graph algorithms to solve problems.
6. Analyze problem constraints and select the most efficient approach.
7. Prepare for coding competitions, technical interviews, and real-world problem solving.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPC 17.1	Identify type of the problem and Apply sorting, searching techniques.	3	3	2	2	3	-	-	1	1	-	2	-	-	L1, L2	
R24CSPC 17.2	Perform bit manipulation and string manipulation operations	3	3	2	2	3	-	-	1	1	-	2	-	-	L2, L3	
R24CSPC 17.3	Compute the given complex problem using mathematical theorem and graph algorithms.	3	3	2	3	3	-	-	1	1	-	2	-	-	L3, L4	
R24CSPC 17.4	Solve given problems using dynamic programming and Greedy Approach	3	3	3	2	3	-	-	1	1	-	3	-	-	L4, L5	
R24CSPC 17	Analyze and develop backtracking algorithms and Geometrical Algorithms.	3	3	3	2	3	-	-	1	1	-	3	-	-	L5, L6	

SYLLABUS**UNIT I:****14 Hours**

Problem Solving Paradigms: The course begins with an introduction, overview, and motivation of competitive programming. It covers problem-solving techniques such as brute force and complete search, and then progresses to efficient algorithmic strategies including divide and conquer, greedy methods, and dynamic programming.

Sorting & Searching: Sorting Theory, Counting Sort, Radix Sort, Ternary Search **CO's–CO1**

Self-Learning Topics: Identify the type of the problem statement. Apply sorting and searching

techniques to solve the given problem.

UNIT II:

15 Hours

Bit Manipulation: Bit representation of integers, basic bit operations (AND, OR, XOR, NOT, left shift, right shift), techniques for bit masking, representing sets using bit manipulation. String terminology and properties, string hashing techniques, pattern matching algorithms, palindrome techniques.

CO's-CO2

Self-Learning Topics: Apply the bit operators to solve given problem. Implement various string operations and pattern matching algorithms

UNIT III:

13 Hours

Number Theory: Primes and factorization, fundamentals of modular arithmetic (modular operations, properties, and applications).

Graphs: Shortest path algorithms – Dijkstra's Algorithm and Bellman-Ford Algorithm. Minimum Spanning Tree: Prim's Algorithm.

Paths and Circuits: Eulerian paths and circuits.

CO's-CO3

Self-Learning Topics: Solve the complex problem using mathematical theorem. Analyzing Graph algorithms to solve suitable problems.

UNIT IV:

13 Hours

Dynamic Programming: Coin change problem, Longest Increasing Subsequence (LIS), Subset Sum problem, counting tiling problems. Greedy Algorithms: Coin change problem (greedy approach), task scheduling with deadlines, minimizing sums (optimal merge patterns / related techniques).

COs-CO4

Self-Learning Topics: Solve complex problem using dynamic programming approach. Solve complex problem using Greedy approach

UNIT V:

13 Hours

Computational Geometry: Points and lines, Polygon (Convex, Concave), Distance functions.

Back tracking: Constructing All Subsets, Constructing All Permutations

CO's-CO5

Self-Learning Topics: Implement Geometrical algorithms. Analyze and develop backtracking algorithms

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Competitive programming 3, by Steven Halim, Felix Halim, Handbook for ACMICPC and IOI contestants.

Reference Books:

1. Data Structures and Algorithms Made Easy by Narasimha Karumachi, Career Monk.
2. Programming challenges by Steven S. Skiena Miguel A. Revilla, The Programming Contest Training Manual, and Springer.
3. Competitive Programmer's Hand book, by Antti Laaksonen, Code Submission Evaluation System

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define brute force technique with an example.
2. What is dividing and conquer? Give one algorithm.
3. Define greedy algorithm. When does it fail?
4. What is time complexity of counting sort?
5. Difference between comparison and non-comparison sorting.
6. What is ternary search? When is it used?
7. Explain divide and conquer with merge sort.
8. Compare brute force vs greedy vs dynamic programming.

L2: Understand

1. Explain bit manipulation techniques with examples.
2. Describe string hashing and its applications.
3. Explain KMP algorithm for pattern matching.
4. What is modular arithmetic?
5. Define a graph and its types.
6. What is the difference between directed and undirected graph?
7. Define shortest path problem.
8. What is a Eulerian path?

L3: Apply

1. Understand the problem statement clearly (input, output, constraints).
2. Identify the correct problem type (sorting, graph, DP, greedy, etc.).
3. Choose an appropriate algorithmic approach (brute force or optimized method).
4. Select suitable data structures (arrays, stacks, queues, trees, hash maps, etc.).
5. Design a step-by-step solution strategy.
6. Implement the solution using a programming language (C++, Java, Python).
7. Optimize the solution to reduce time and space complexity.
8. Handle edge cases and special test conditions.
9. Debug and verify correctness of the program.
10. Practice regularly through coding platforms and contests.

L4: Analysing

1. Compare brute force, greedy, and dynamic programming approaches. When should each be

2. Used?
3. Analyze and justify when counting sort or radix sort is better than comparison-based sorting.
4. Explain the advantages of bit manipulation. When is bitmasking more efficient than arrays?
5. Compare naive pattern matching and KMP algorithm with respect to time complexity and usage.
6. Analyze string hashing. What are collisions and how can they be avoided?
7. Compare Dijkstra's algorithm and Bellman-Ford algorithm. When should each be used?
8. Explain why Dijkstra's algorithm fails for graphs with negative edge weights.
9. Analyze Prim's algorithm for minimum spanning tree. When is it most efficient?
10. Compare dynamic programming and greedy algorithms with suitable examples.
11. Analyze the Longest Increasing Subsequence problem. Compare $O(n^2)$ and $O(n \log n)$ solutions.
12. Explain the counting tiling problem and derive its recurrence relation.
13. Compare backtracking and brute force. When does backtracking become inefficient?

L5: Evaluating

1. Check correctness of output for all test cases.
2. Analyze time complexity (Big-O notation).
3. Analyze space complexity (memory usage).
4. Compare different approaches for efficiency.
5. Test edge cases and boundary conditions.
6. Verify performance for large inputs.
7. Ensure optimal algorithm selection.
8. Validate final output with expected results.
9. Review code clarity and structure.
10. Confirm solution meets problem constraints.

L6: Create

1. Understand the problem statement clearly.
2. Identify inputs, outputs, and constraints.
3. Break the problem into smaller sub problems.
4. Choose the appropriate algorithmic approach.
5. Select suitable data structures.
6. Design a step-by-step solution plan.
7. Convert the plan into an algorithm.
8. Implement the solution in code.
9. Test with sample and edge cases.
10. Optimize for time and space efficiency.

**Chairperson
Board of Studies (CSE)**

R24CSPE01 INFORMATION RETRIEVAL AND SEARCH ENGINES 3 0 0 3
 (Professional Elective-I)
Course Objectives:

1. Understand the fundamentals of information retrieval using Boolean models, including inverted index construction, query processing, and text preprocessing techniques such as tokenization, stemming, and phrase indexing.
2. Understand dictionary structures and indexing techniques in information retrieval systems, along with tolerant retrieval methods such as wildcard queries, spelling correction, and efficient index construction methods.
3. Understand statistical properties of terms in documents and apply compression techniques for indexes, along with term weighting methods such as TF-IDF for effective information retrieval.
4. Understand the vector space model for document scoring and ranking, including query representation, similarity computation, and document weighting schemes.
5. Understand evaluation techniques for information retrieval systems and gain knowledge of web search engine concepts, including crawling, indexing, and web structure analysis.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSP E01.1	Describe the pre-processing methods for effective information retrieval	3	2	--	--	2	--	--	--	--	--	--	3	2	L1, L2	
R24CSP E01.2	Explain about tolerant information retrieval	3	2	1	--	2	--	--	--	--	--	--	3	2	L2, L3	
R24CSP E01.3	Illustrate the index compression process	3	3	2	1	2	--	--	--	--	--	--	--	--	L3, L4	
R24CSP E01.4	Build textual information into vectors	3	3	3	2	3	--	--	--	--	--	--	3	3	L4, L5	
R24CSP E01.5	Make use of a web search engine for crawling the data	3	3	3	2	3	1	--	--	--	2	--	3	3	L5, L6	

SYLLABUS**UNIT I:****15 Hours**

BOOLEAN RETRIEVAL: An example information retrieval problem, a first take at building an inverted index, processing Boolean queries, the extended Boolean model versus ranked retrieval.

THE TERM VOCABULARY AND POSTINGS LISTS: Document delineation and character sequence decoding, obtaining the character sequence in a document, choosing a document unit, Determining the vocabulary of terms, Tokenization, dropping common terms: stop words, Normalization (equivalence class in gof terms) stemming and lemmatization, Faster postings list

intersection via skip pointers, Positional postings and phrase queries, Bi-word indexes, Positional indexes, Combination schemes. **CO's–CO1**

Self-Learning Topics: Lemmatization using NLP Tools, Real-Time Inverted Index Construction

UNIT II:

13 Hours

DICTIONARIES AND TO LERAN TRETRIEVAL: Search structures for dictionaries, Wild card queries, General wild card queries, k-gram indexes for wild card queries, spelling correction, implementing spelling correction, Forms of spelling correction, Edit distance, k-gram indexes for spelling correction, Context sensitive spelling correction, Phonetic correction.

INDEXCONSTRUCTION: Hardware basics, Blocked sort-based indexing, Single-passin-memory indexing, distributed indexing, dynamic indexing, other types of indexes **CO's–CO2**

Self-Learning Topics: Trie Data Structure for Dictionary Search, Search Auto-Completion Techniques

UNIT III:

15 Hours

INDEX COMPRESSION: Statistical properties of terms in information retrieval, Heaps' law: Estimating the number of terms, Zipf's law: Modeling the distribution of terms, Dictionary compression, Dictionary as a string, Blocked storage, Postings file compression, Variable byte codes, γ -codes. **SCORING, TERM WEIGHTING:** Parametric and zone indexes, Weighted zone scoring, Learning weights, The optimal weight g, Term frequency and weighting, Inverse document frequency, Tf-idf weighting. **CO's–CO3**

Self-Learning Topics: Compression Algorithms in Search Engines, Machine Learning for Weight Optimization

UNIT IV:

13 Hours

THE VECTOR SPACE MODEL: The vector space model for scoring, Dot products, Queries as vectors, Computing vector scores, Variant tf-Idf functions, Sub linear scaling, maximum Formalization, Document and query weighting schemes, Pivoted normalized document length

COs–CO4

Self-Learning Topics: AI-Based Search Ranking Systems, Word Embeddings in Information Retrieval

UNIT V:

13 Hours

EVALUATIONININ FORMATION RETRIEVAL Information retrieval system evaluation, Standard test collections, Evaluation of unranked retrieval sets, Evaluation of ranked retrieval results.

WEB SEARCH ENGINE: Web search basics, Web characteristics, Index size and estimation, Near-duplicates and shingling, Web crawling and indexes, Distributing indexes, Connectivity servers. **CO's–CO5**

Self-Learning Topics: Basics of Google Search Algorithm, Ethical Issues in Web Search and Data Crawling

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

Modern Information Retrieval Systems and Web Search Engines (Industry practices in search engines, large-scale indexing, web crawling, ranking algorithms, TF-IDF and vector space model applications in systems like Google Search, Elastic search, and enterprise search platforms)

Text Books:

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *An Introduction to Information Retrieval*, 1st Edition, Cambridge University Press, 2008.

Reference Books:

1. G G Chowdhury “Introduction to Modern Information Retrieval” ,3rd Edition, neal-schuman publishers, 2010.
2. Gerald J. Kowalski, Mark T.Maybury, “Information storage and Retrieval systems: theory and implementation”, 2nd Edition, kluwer academic publishers, 2009.

Web References:

1. NPTEL (SWAYAM):<https://nptel.ac.in>
2. <http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Information Retrieval system. (Unit I)
2. What is an inverted index? (Unit I)
3. Define tokenization. (Unit I)
4. What is wildcard query? (Unit II)
5. Define TF-IDF. (Unit III)

L2: Understand

1. Explain Boolean retrieval model. (Unit I)
2. Describe tokenization and normalization in text processing. (Unit I)
3. Explain k-gram index in wildcard queries. (Unit II)
4. Describe edit distance in spelling correction. (Unit II)
5. Explain vector space model. (Unit IV)

L3: Apply

1. Apply inverted index to process Boolean queries. (Unit I)
2. Use stemming and lemmatization for text preprocessing. (Unit I)
3. Apply k-gram indexing for wildcard query matching. (Unit II)
4. Compute TF-IDF values for a document set. (Unit III)
5. Apply vector space model to rank documents for a query. (Unit IV)

L4: Analysing

1. Analyze Boolean retrieval vs ranked retrieval models. (Unit I)
2. Compare stemming and lemmatization techniques. (Unit I)
3. Analyze different forms of spelling correction methods. (Unit II)
4. Compare variable byte codes and gamma codes for compression. (Unit III)
5. Analyze cosine similarity in vector space model. (Unit IV)

L5: Evaluating

1. Evaluate the effectiveness of inverted index in search systems. (Unit I)
2. Assess wildcard and spelling correction techniques in search accuracy. (Unit II)
3. Evaluate TF-IDF weighting in information retrieval systems. (Unit III)
4. Assess the advantages of vector space model over Boolean model. (Unit IV)
5. Evaluate web search engine indexing methods. (Unit V)

L6: Create

1. Design an inverted index system for document retrieval. (Unit I)
2. Develop a spelling correction system using edit distance. (Unit II)
3. Design a compression scheme for posting lists. (Unit III)
4. Create a document ranking system using vector space model. (Unit IV)
5. Develop basic web search engine architecture. (Unit V)

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To familiarize the concepts and architectural types of data Warehouses.
2. Provides efficient design and management of data storages using data warehousing and OLAP
3. To understand the fundamental processes concepts and techniques of data mining.
4. To consistently apply knowledge concerns current data mining research and how this may contribute to the effective design and implementation of data mining applications.
5. To provide advance research skills through the investigation of data- mining Literature

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPE 01.1	Design a data mart or data warehouse for any organization.	3	3	2	2	2	-	-	-	-	-	-	3	2	L1, L2
R24CSPE 01.2	Apply Association and classification knowledge to different data sets.	3	2	2	2	3	-	-	-	-	1	-	3	3	L2, L3
R24CSPE 01.3	Apply the clustering Techniques for different data sets.	3	3	2	2	3	-	-	-	-	1	-	3	3	L3, L4
R24CSPE 01.4	Identify the similar objects using clustering techniques.	2	2	2	2	3	-	-	-	-	1	2	2	3	L4, L5
R24CSPE 01.5	Explore recent trends in data mining such as web mining, spatial-temporal mining.	2	2	1	2	2	-	-	-	-	1	2	2	2	L5, L6

SYLLABUS**UNIT –I****15 Hours**

Data Warehouse and OLAP Technology: what is a Data Warehouse, Multidimensional Data Model, OLAP Operations on Multidimensional Data, Data Warehouse Architecture Cube computation: Multiway Array Aggregation, BUC

CO's -CO1

Self-Learning Topics: Advanced cube computation techniques, OLAP vs. OLTP in practical business scenarios

UNIT - II**13 Hours**

Introduction to Data Mining: Fundamentals of data mining, KDD, Challenges, Data Mining Functionalities, Data Mining Task Primitives, Major issues in Data Mining. Data Preprocessing: Needs for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction

CO'S - CO2

Self-Learning Topics: Data quality issues in real-world datasets, Ethical issues in data mining

UNIT - III

13 Hours

Mining Frequent Pattern: Associations and Correlations: Basic Concepts, Efficient and Scalable Frequent Item Set Generation, The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIORI Algorithm, The Partition Algorithms, FP Growth Algorithms, Compact Representation of Frequent Item Set- Maximal Frequent Item Set, Closed Frequent Item Set. Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification

Self-Learning Topics: Improvements over APRIORI (ECLAT, FP-Growth variations), Over fitting and under fitting problems

CO's -CO3

UNIT - IV

15 Hours

Cluster Analysis: Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods- K- means, PAM, Hierarchical Methods- BIRCH, Density- Based Methods- DBSCAN, and Outlier Detection

Self-Learning Topics: Advanced clustering methods (OPTICS, Mean-Shift), Applications of clustering in image processing

CO's -CO4

UNIT-V:

14 Hours

Overview Sequential pattern mining: Mining time series data, mining sequence patterns in transactional databases, graph mining, spatial data mining, text mining, mining the World Wide Web. Data Mining Applications

CO's-CO5

Self-Learning Topics: Data mining in AI and machine learning pipelines, social network analyses

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, Second Edition, Elsevier, 2007.
2. Data Mining – Vikram Pudi & P.Radha Krishna-Oxford Publication.

Reference Books:

1. Data Mining Concepts and Techniques, Han, Morgan Kaufmann publishers.
2. Introduction to Data Mining by Pang –Ning Tan, Vipin Kumar, Michael Steinbach

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Data Warehouse.
2. What is a Data Cube?
3. List the OLAP operations.
4. Define Data Mining.
5. What is KDD (Knowledge Discovery in Databases)?
6. List any four Data Preprocessing techniques.
7. Define Support and Confidence.
8. What is the APRIORI Principle?
9. Define Clustering.
10. Name different types of Clustering Methods.
11. What is Sequential Pattern Mining?

L2: Understand

1. Explain the characteristics of a Data Warehouse and how it differs from operational databases.
2. Explain the Multidimensional Data Model with a suitable example (dimensions, measures, and cube).
3. Describe different OLAP operations with examples (roll-up, drill-down, slice, dice, and pivot).
4. Explain the KDD process and its steps in detail.
5. Discuss the major challenges in Data Mining.
6. Explain how association rules are generated using support and confidence.
7. Describe the working of the APRIORI algorithm with an example.
8. Explain the K-means clustering algorithm with steps and example.
9. Compare K-means and DBSCAN clustering methods.
10. Explain Sequential Pattern Mining with an example.
11. Discuss different Data Mining applications in real-world scenario

L3: Apply

1. Given a sales dataset (Product, Time, Location), construct a data cube and demonstrate any two OLAP operations (slice, dice, roll-up, drill-down).
2. Design a simple Data Warehouse schema (star or snowflake) for a retail store and identify fact and dimension tables.
3. Given a dataset with missing and noisy values, apply suitable data cleaning techniques.
4. Perform data transformation and normalization on a given dataset.
5. Given a transaction dataset, apply the APRIORI algorithm to find frequent item sets (show all steps).
6. Generate association rules from frequent item sets and calculate support and confidence.
7. Construct a Decision Tree for a given dataset using appropriate splitting criteria.
8. Apply the K-means algorithm on a given set of data points (show iterations and cluster formation).
9. Given a dataset, identify clusters using DBSCAN (core, border, noise points).
10. Given a sequence database, identify sequential patterns.
11. Analyze a real-world dataset and suggest suitable data mining techniques (classification, clustering, and association) with justification.

4: Analysing

1. Analyze the differences between Data Warehouse and OLTP systems in terms of structure, usage, and performance.
2. Compare different OLAP operations and analyze when each operation is most suitable.
3. Analyze the impact of poor data quality on data mining results and decision-making.
4. Compare different data preprocessing techniques and evaluate when each should be applied.
5. Compare APRIORI and FP-Growth algorithms in terms of efficiency, scalability, and limitations.
6. Analyze the trade-off between support and confidence in association rule mining.
7. Compare Decision Tree and Bayesian Classification methods and analyze their strengths and weaknesses.
8. Analyze the differences between partitioning methods (K-means) and density-based methods (DBSCAN).
9. Analyze the challenges involved in **mining time-series and sequential data**.
10. Compare different data mining applications (e.g., web mining, text mining, spatial mining) and analyze their unique requirements.
11. Evaluate the limitations of K-means clustering and suggest possible improvements.

L5: Evaluating

1. Evaluate the importance of Data Warehouses in decision support systems. Are they always necessary? Justify your answer.
2. Evaluate different Data Warehouse architectures and justify which is most suitable for large-scale organizations.
3. Evaluate the effectiveness of different data preprocessing techniques in improving data mining results.
4. “Data preprocessing is the most critical step in data mining.” Evaluate this statement with justification.
5. Evaluate the performance of APRIORI vs. FP-Growth for large datasets. Which would you recommend and why?
6. Evaluate different classification techniques (Decision Tree vs. Bayesian). Which is better under different conditions?
7. Evaluate the suitability of K-means clustering for real-world datasets with noise and outliers.
8. “DBSCAN is better than K-means for cluster analysis.” Evaluate this statement with proper reasoning.
9. Evaluate the challenges and effectiveness of Sequential Pattern Mining in real-world applications.
10. Evaluate different data mining applications (web, text, spatial) and justify which has the highest impact today.
11. Assess the role of Data Mining in business decision-making. Is it always reliable? Justify your answer.

L6: Create

1. Design Data Warehouse architecture for an e-commerce system, including ETL process, data sources, and data marts.
2. Create a multidimensional data model (data cube) for a hospital management system showing relevant dimensions and measures.

3. Design a complete data preprocessing pipeline for a real-world dataset (include cleaning, integration, transformation, reduction).
4. Develop a data mining solution for detecting fraudulent transactions in banking systems.
5. Design an improved version of the APRIORI algorithm to handle large-scale datasets efficiently.
6. Create a classification model for predicting student performance using suitable techniques (Decision Tree / Bayesian).
7. Design a clustering approach for customer segmentation in a retail business.
8. Develop a method to detect outliers in a large dataset using clustering or statistical techniques.
9. Design a system for Sequential Pattern Mining in an online shopping platform.
10. Create a web mining framework to analyze user behavior on websites.
11. Develop a data mining application for a real-world problem (healthcare, education, agriculture, etc.) with architecture and methodology.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. Understand the fundamental concepts, principles, and scope of Human–Computer Interaction and its role in designing interactive systems.
2. Learn usability goals, human factors, and user-centered design (UCD) principles for developing effective interfaces.
3. Analyze different interaction styles such as menus, forms, command languages, and natural language interfaces.
4. Design and evaluate user interfaces based on usability principles, visual design guidelines, and user experience metrics.
5. Understand interaction devices and modern input/output technologies used in computing systems.
6. Explore information visualization, search systems, and emerging technologies in HCI such as intelligent and immersive interfaces.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSP E03.1	Understand HCI fundamentals, usability principles, accessibility, and design processes.	3	2	1	1	1	1	1	-	1	-	2	2	1	L1, L2	
R24CSP E03.2	Apply interaction design principles to develop effective and user-friendly interfaces.	2	3	3	2	3	1	1	1	2	1	2	3	3	L3	
R24CSP E03.3	Analyze different interaction modalities, devices, and interface types.	2	3	2	2	2	1	-	-	1	-	2	3	2	L3, L4	
R24CSP E03.4	Evaluate user experience based on performance, usability, and design principles.	2	3	2	3	2	2	2	-	1	-	2	2	2	L4	
R24CSP E03.5	Design user-centered systems, documentation, and information visualization solutions.	2	2	3	2	3	2	1	1	2	2	2	3	3	L5, L6	

SYLLABUS**UNIT I: Foundations & Design Processes****12 Hours**

Introduction to Human–Computer Interaction, usability of interactive systems, usability goals and measures, usability motivations, universal usability and accessibility standards, organizational

design to support usability, User-Centered Design (UCD) lifecycle, design thinking, ethnographic observation, participatory design, scenario development, user journeys, ethical, social and legal issues in HCI, basic usability testing methods. **CO's- CO1**

Self-Learning Topics: Agile UX methodologies, remote usability testing, UX analytics tools.

UNIT II: Interaction Design & Modern Interfaces

12 Hours

Menu selection principles, task-related menu organization, content organization, form fill-in, dialog boxes and alternatives, mobile-first design, responsive design, conversational user interfaces, gesture-based interaction, micro interactions and feedback mechanisms, design for small displays such as smart phones and wearable devices. **CO's-CO2**

Self-Learning Topics: chat bot and voice-based interfaces, wearable device UI case studies.

UNIT III: Interaction Modalities & Devices

10 Hours

Command-based interfaces, graphical user interfaces, natural language interfaces, interaction devices including keyboards, keypads, pointing devices, touch interfaces, stylus, speech and auditory interfaces, displays including small and large screens.

Self-Learning Topics: multimodal interaction, eye-tracking systems, gesture recognition systems, introduction to Augmented Reality and Virtual Reality interfaces. **CO's-CO3**

UNIT IV: UX Design, Performance & Evaluation

12 Hours

Quality of service in user experience, models of response-time impact, user expectations and attitudes, user productivity, variability in response time, frustrating experiences, error messages and feedback systems, display design, web page design, window design, colour theory, UX evaluation basics. **CO's-CO4**

Self-Learning Topics: UX analytics and heatmaps, personalization using Artificial Intelligence, ethical UX and dark pattern case studies.

UNIT V: Information Systems & Visualization

12 Hours

User documentation and online help systems, online tutorials, development process, information search in textual documents, database querying, multimedia document search, basic filtering techniques, introduction to information visualization and dashboards.

Self-Learning Topics: recommendation systems, semantic search techniques, big data visualization tools. **CO's-CO5**

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. The Essential guide to user interface design,2/e, Wilbert O Galitz, Wiley Dreama Tech.
- Operating Systems- A Concept based Approach, D. M. Dhamdhare, 2nd Edition, Tata McGraw-Hill
2. Designing the User Interface, Strategies for Effective Human Computer Interaction, 5ed, Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M Jacobs, Pearson

Reference Books:

1. Human Computer, Interaction Dan R.Olsan, Cengage, 2010.
2. Designing the user interface. 4/e, Ben Shneidermann, PEA.
3. User Interface Design, Soren Lauesen , PEA
4. User Interface Design, Soren Lauesen , PEA

Web References:

1. https://onlinecourses.nptel.ac.in/noc26_cs70/

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define Human–Computer Interaction (HCI).
2. What are usability goals and measures?
3. Define universal usability.
4. What is User-Centered Design (UCD)?
5. Define menu selection.
6. What is form fill-in?
7. Define dialog boxes.
8. What are command-based interfaces?
9. Define graphical user interfaces (GUI).
10. What are natural language interfaces?
11. List different interaction devices used in HCI.
12. Define quality of service in user experience.
13. What are error messages?
14. Define user documentation.

L2: Understand

1. Explain usability motivations with examples.
2. Describe the UCD lifecycle.
3. Explain accessibility standards in design.
4. Describe principles of content organization.
5. Explain dialog boxes with examples.
6. Describe mobile-first and responsive design.
7. Explain speech and auditory interfaces.
8. Describe touch-based interaction.
9. Explain response-time impact on users.
10. Describe user expectations in UX design.

11. Explain display design principles.
12. Describe color theory in UI design.
13. Explain online help systems

L3: Apply

1. Apply design thinking to create a simple UI.
2. Illustrate ethnographic observation in system design.
3. Apply menu organization principles to a system.
4. Design a form fill-in interface for registration.
5. Apply mobile-first design to an application.
6. Demonstrate the use of pointing devices in GUI.
7. Apply natural language interaction in a system.
8. Improve a website using UX principles.
9. Apply filtering techniques in search systems.
10. Design a simple online help system.

L4: Analysing

1. Analyze the role of organizational design in usability.
2. Compare participatory design and scenario development.
3. Analyze menu-based vs dialog-based interfaces.
4. Compare mobile and desktop interface design.
5. Analyze advantages of touch interfaces.
6. Compare command-line and GUI interfaces.
7. Analyze response time vs user productivity.
8. Compare different feedback mechanisms in UI.
9. Analyze textual vs multimedia search systems.
10. Compare database querying and document searching.

L5: Evaluating

1. Evaluate ethical and legal issues in HCI.
2. Assess usability testing methods.
3. Evaluate effectiveness of mobile UI design.
4. Assess feedback mechanisms in interfaces.
5. Evaluate different interaction devices.
6. Assess impact of display size on usability.
7. Evaluate web page design based on UX principles.
8. Assess color usage in interface design.
9. Evaluate online documentation systems.
10. Assess effectiveness of dashboards.

L6: Create

1. Design a user journey for an online system.
2. Create a usability testing plan.
3. Design a responsive web page layout.
4. Create a menu structure for an application.
5. Design a graphical user interface for a text editor.
6. Create a natural language-based system concept.
7. Design effective error messages.
8. Create a user-friendly web page.

9. Design a dashboard for data visualization.
10. Create a structured online help guide

**Chairperson
Board of Studies (CSE)**

R24CSPE04**NoSQL DATABASES****3 0 0 3**

(Professional Elective-I)

Course Objectives:

1. To understand the basics of NoSQL databases, their need, types, and differences from traditional databases.
2. To learn how NoSQL databases work in distributed environments, including concepts like replication, sharding, and consistency.
3. To develop basic skills in using MongoDB for storing, retrieving, and managing data.
4. To explore column-family and key-value databases such as Cassandra, HBase, and Redis and their real-world uses.
5. To understand graph databases like Neo4j and learn about modern trends such as cloud NoSQL and data security.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		P01	P02	P03	P04	P05	P06	P07	P08	P09	P010	P011	PSO1	PSO2		
R24CSPE04.1	Understand fundamentals and different data models of NoSQL databases.	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2	
R24CSPE04.2	Understand distribution, consistency, and scalability in NoSQL systems.	3	3	-	2	3	-	-	2	2	-	2	3	2	L2, L3	
R24CSPE04.3	Apply document database concepts using MongoDB.	3	3	-	2	3	-	-	-	-	-	2	3	3	L3, L4	
R24CSPE04.4	Analyze column-family and key-value databases.	3	-	-	2	3	-	-	-	-	-	2	3	2	L4, L5	
R24CSPE04.5	Apply graph and modern NoSQL databases in real-world applications.	3	2	3	2	3	2	3	-	-	2	3	-	-	L5, L6	

SYLLABUS**UNIT-I: Introduction to NoSQL & Data Models****15 Hours**

Overview and Evolution of databases and need for NoSQL, Limitations of relational databases (scalability, schema rigidity), Impedance mismatch and Big Data challenges, Characteristics of NoSQL databases, Types of NoSQL databases, Key-Value model, Document model, Column-Family model, Graph model, Aggregate-oriented databases, Comparison of RDBMS vs NoSQL, Overview of popular NoSQL databases (MongoDB, Cassandra, HBase, Neo4j).

COs-CO1

Self-Learning Points: Comparison of RDBMS vs NoSQL, Overview of popular NoSQL databases (MongoDB, Cassandra, HBase, Neo4j)

UNIT-II: Distributed NoSQL Systems**14 Hours**

Distributed database concepts, Data distribution models (single node, sharding), Replication

techniques (master-slave, peer-to-peer), CAP theorem (consistency, availability, partition tolerance), BASE properties vs ACID properties, Consistency models (strong, eventual consistency), Transactions in NoSQL, Fault tolerance and high availability, Introduction to MapReduce, Basics of distributed processing (Hadoop overview) **COs-CO2**

Self-Learning Topics: Transactions in NoSQL, Introduction to MapReduce, and Basics of distributed processing (Hadoop overview).

UNIT-III: Document Databases (MongoDB)

13 Hours

Introduction to document databases, MongoDB architecture and components, CRUD operations in MongoDB, Indexing and querying, Aggregation framework, Schema design and data modelling, Replication and sharding in MongoDB, Integration with applications (basic overview), Use cases (content management, e-commerce, analytics), Limitations and best practices. **COs-CO3**

Self-Learning Topics: Event Logging, Blogging Platforms, Counters, Expiring Usage, When Not to Use.

UNIT-IV: Column-Family & Key-Value Databases

12 Hours

Column-family databases overview, Apache HBase architecture and data model, Apache Cassandra architecture and features, Distributed storage and scalability, Key-value databases overview, Redis/Riak concepts and features, In-memory databases and caching, Performance and optimization techniques, Use cases (event logging, sessions, IoT data). **COs-CO4**

Self-Learning Topics: Storing Session Information, User Profiles, Preferences, Shopping Cart Data, When Not to Use.

UNIT-V: Graph Databases & Modern Trends

12 Hours

Graph databases overview, Neo4j architecture and data model, Nodes, relationships, and properties, Query language (Cypher basics), Graph traversal and analytics, Use cases (social networks, recommendation systems, fraud detection), multi-model databases overview, Cloud NoSQL databases (AWS DynamoDB, Google Firestore – basics), Security and data privacy in NoSQL, Future trends in NoSQL. **COs-CO5**

Self-Learning Topics: Connected Data, Routing, Dispatch, and Location-Based Services, Recommendation Engines.

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Pramod J. Sadalage, Martin Fowler, Pearson Education, 2013.
2. Shashank Tiwari. Professional NoSQL. John Wiley and Sons. ISBN:978-0-470-94224-6.

Reference Books:

1. A Guide to Modern Databases and the NoSQL Movement Edition, Redmond, E.& Wilson Redmond, E. & Wilson, J. (2012).
2. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (1st Ed.). Raleigh, NC: The Pragmatic Programmers, LLC.
3. Dan Sullivan. NoSQL for Mere Mortals. Addison-Wesley Professional.2015.
4. Guy Harrison. Next-Generation Databases. Apress.2016.

Web References:

1. [Introduction to NoSQL Databases \(Coursera\)](#)
2. [Free NoSQL Course: Master Non-Relational Databases](#)

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define NoSQL database.
2. List types of NoSQL databases.
3. What is CAP theorem?
4. Define sharding.
5. What is MongoDB?
6. Define graph database.
7. Explain evolution of databases and need for NoSQL.
8. Describe characteristics of NoSQL databases.
9. Explain CAP theorem with components.
10. Describe MongoDB architecture.
11. Explain Cassandra architecture.
12. Describe Neo4j architecture and components.

L2: Understand

1. What are limitations of RDBMS?
2. Define impedance mismatch.
3. What is BASE property?
4. Define replication.
5. What is indexing in MongoDB?
6. Define key-value store.
7. Explain NoSQL data models with examples.
8. Compare RDBMS and NoSQL databases.
9. Explain ACID vs BASE properties.
10. Describe consistency models in distributed systems.
11. Explain CRUD operations in MongoDB.
12. Describe column-family databases with examples.

L3: Apply

1. Identify suitable NoSQL model for JSON data.
2. Which database is used for session storage?
3. Identify type of replication for high availability.
4. Write a MongoDB insert query.
5. Identify use of Redis.
6. Write a simple Cypher query.
7. Apply key-value model for session management.
8. Demonstrate document model with example.
9. Apply sharding for distributed databases.
10. Demonstrate CRUD operations in MongoDB.
11. Apply Cassandra for scalable data storage.
12. Apply graph database for social networks.

L4: Analysing

1. Differentiate RDBMS and NoSQL.
2. Compare document and column-family models.
3. Differentiate strong and eventual consistency.
4. Compare master-slave and peer-to-peer replication.
5. Compare indexing and aggregation.
6. Compare graph and relational databases.
7. Analyze NoSQL vs relational databases.
8. Analyze CAP theorem trade-offs.
9. Analyze distributed database challenges.
10. Analyze MongoDB performance using indexing.
11. Compare HBase and Cassandra architectures.
12. Analyze graph traversal techniques.

L5: Evaluating

1. Why use NoSQL for Big Data?
2. Justify schema-less design.
3. Why is replication important?
4. Evaluate eventual consistency.
5. Why MongoDB for real-time apps?
6. Evaluate graph databases usage.
7. Evaluate advantages of NoSQL databases.
8. Critically evaluate NoSQL limitations.
9. Evaluate distributed systems using CAP theorem.
10. Evaluate MongoDB for large-scale applications.
11. Evaluate Cassandra for distributed systems.
12. Evaluate Neo4j for recommendation systems.

L6: Create

1. Design key-value model for user sessions.
2. Create document schema for user data.
3. Design replication strategy.
4. Create MongoDB collection structure.
5. Design key-value cache system.

6. Create graph model for social network.
7. Design NoSQL solution for e-commerce system.
8. Develop document schema for blogging platform.
9. Design distributed NoSQL system with sharding.
10. Develop MongoDB schema for real-time applications.
11. Design column-family database for IoT data.
12. Develop graph database for recommendation system.

**Chairperson
Board of Studies (CSE)**

R24CSOE01**DATA STRUCTURES****3 0 0 3**

(Open Elective -1)

Course Objectives:

1. To introduce fundamental concepts of data structures and algorithms.
2. Be familiar with basic techniques of algorithm analysis.
3. Solve problems using data structures such as linear lists, stacks, queues, hash tables.
4. Explore basic concepts of hashing and apply it to solve problems requiring fast data retrieval and management.
5. To understand real-world applications of data structures.
6. Be familiar with advanced data structures such as balanced search trees, AVL Trees, and B Trees.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Do k
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE 01.1	Explain the role of data structures in organizing and data efficiently in algorithms	3	2	2	2	-	-	-	-	-	-	-	3	2	L1, L2	
R24CSOE 01.2	Develop programs using stacks and Queues to solve related problems.	3	3	2	2	2	-	-	-	-	-	-	3	2	L2, L3	
R24CSOE 01.3	Use linked lists effectively for dynamic data handling.	3	3	2	2	2	-	-	-	-	-	-	3	3	L3, L4	
R24CSOE 01.4	Demonstrate the operations such as Insertion, Deletion and Search on Data structures like, AVL trees and B Trees.	3	2	2	2	3	-	-	-	-	-	-	3	2	L4, L5	
R24CSOE 01.5	Apply trees and graphs in real-world scenarios.	3	2	3	2	3	-	-	-	-	-	-	-	-	L5, L6	

SYLLABUS**UNIT-I:****14 Hours**

Introduction to Linear Data Structures: Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity.

Searching Techniques: Linear & Binary Search.

COs–CO1

Sorting Techniques: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Merge Sort.

Self-Learning Topics: Comparison of sorting algorithms with real-time examples.

UNIT II: Stacks, Queues**15 Hours**

Stack Abstract Data Type, Queue Abstract Data Type, Stacks and Queues using arrays, Introduction to Evaluation of Expressions, Evaluating Postfix Expressions, Infix to Postfix and Prefix conversion, Circular Queues using arrays, Dynamically Allocated Storage using pointers.

COs–CO2

Self-Learning Topics: Applications of stack:(Parenthesis checking, Undo/Redo operations)

UNIT III:**15 Hours**

Singly Linked Lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list.

Circular Linked Lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from Circular Linked Lists

Doubly Linked Lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from Doubly Linked Lists

COs–CO3

Self-Learning Topics: Skip Lists

UNIT-IV:**13 Hours**

Trees: Introduction to Trees, Properties of Binary Tree, Representation of Binary Tree using Arrays and Linked List, Binary Search Tree –Insertion, Deletion & Traversals, BTrees, B+ Trees. AVL Trees.

COs–CO4

Self-Learning Topics: Segment Trees and Fenwick Trees

UNIT V:**15 Hours**

Graphs: Graph Abstract Data Type, Definitions, Graph Representations, Elementary Graph Operations, Depth First Search, Breadth First Search, Spanning Trees, Minimum Cost Spanning Trees, Prim's and Kruskal's Algorithms, Shortest Paths and Transitive Closure, Single Source All Destination - Dijkstra's Algorithm

COs–CO5

Self-Learning Topics: Graph coloring problems

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

Cos**POs / PSOs**

1. Graphs Algorithm

CO5**PO1, PO2, PO3****Text Books:**

1. Fundamentals of Data Structures in C, 2nd edition, Horowitz, Sahani and Anderson-Freed, Universities Press, 2008.
2. 'Data Structures and Algorithm Analysis in C' by Mark Allen Weiss, Pearson.
3. 'Data Structures Using C' by Reema Thareja, Oxford University Press.

Reference Books:

1. Algorithms, Part I and II' by Robert Sedgewick and Kevin Wayne, Addison-Wesley.
2. Data Structures and Algorithms Made Easy' by Narasimha Karumanchi, Career Monk Publications.
3. Data Structures: A PseudoCode Approach with C, 2nd Edition, Richard F. Gilberg, & Behrouz A. Forouzon, Cengage

Web References:

1. <http://www.hackerrank.com/domains/datastructures>
2. http://www.github.com/topics/data_structures_c
3. <http://nptel.ac.in/courses>
4. <http://www.cslibrary.stanford.edu>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define linear data structure with examples.
2. What is an Abstract Data Type (ADT)?
3. Define time complexity and space complexity.
4. State the difference between linear and non-linear data structures.
5. What is Binary Search? When is it applicable?
6. Define stack and its basic operations.

L2: Understand

1. What is stack overflow and underflow?
2. Explain stack implementation using arrays and linked lists.
3. Write an algorithm to convert infix to postfix expression.
4. Difference between singly and circular linked lists. List advantages of linked lists over arrays.
5. Define tree and binary tree.
6. What is a Binary Search Tree (BST)?
7. List types of tree traversals.
8. What are DFS and BFS?

L3: Apply

1. Write the algorithm for Linear Search.
2. Trace Binary Search for the element 25 in the list [10, 15, 20, 25, 30].
3. Convert the infix expression $(A+B)C$ into postfix.
4. Evaluate a postfix expression: $2354*+9-$
5. Write an algorithm to convert infix to postfix expression.
6. Perform enqueue and dequeue operations on a circular queue.
7. Write steps to insert a node at the beginning.
8. Write algorithms for BST insertion and deletion.

L4: Analysing

1. Compare all sorting techniques based on time complexity and stability.
2. Explain how stacks are used in backtracking.
3. Explain collision in hashing.
4. Analyze hashing techniques and collision resolution methods.

5. Compare arrays and linked lists with advantages and limitations.
6. Compare adjacency matrix and adjacency list.
7. Compare DFS and BFS with examples.

L5: Evaluating

1. Which sorting algorithm is best for small datasets? Justify your answer.
2. Compare stack applications in recursion vs. iteration.
3. Analyze hashing techniques and collision resolution methods.
4. Analyze linked list applications in polynomial representation.
5. Analyze the advantages of AVL trees over BST.

L6: Create

1. Design an efficient sorting approach for a real-time application (e.g., ranking students).
2. Design a program-using stack to check balanced parentheses.
3. Design a scheduling system using priority queue.
4. Design a method to swap two nodes without swapping data.
5. Design a graph traversal algorithm for shortest path finding.

**Chairperson
Board of Studies (CSE)**

R24CSOE02**DATABASE MANAGEMENT SYSTEMS****3 0 0 3**

(Open Elective-I)

Course Objectives:

1. To understand the fundamental concepts of database systems, data models, and database architecture.
2. To design and represent real-world problems using ER diagrams and convert them into relational schemas.
3. To apply relational model concepts and write SQL queries for data definition, manipulation, and retrieval.
4. To analyze database design using normalization techniques to reduce redundancy and ensure data integrity.
5. To understand transaction management, concurrency control, recovery techniques, and indexing methods for efficient database operations.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSOE 02.1	Explain characteristics, architecture, data models, schemas, and components of DBMS.	3	-	-	2	2	-	-	-	-	-	2			L1, L2
R24CSOE 02.2	Design ER diagrams and transform them into relational schemas using constraints and inheritance concepts.	3	3	2	2	3	-	-	-	-	-	2	3	3	L2, L3
R24CSOE 02.3	Apply relational algebra, relational calculus, and SQL queries for database creation and manipulation.	3	3	3	2	3	-	-	-	-	-	2	3	3	L3, L4
R24CSOE 02.4	Analyze and normalize database schemas using functional dependencies and normal forms.	3	3	2	3	3	-	-	-	-	-	2	3	2	L4, L5
R24CSOE 02.5	Evaluate transaction management, concurrency control, serializability, deadlock handling, and recovery techniques.	3	3	2	3	3	2	-	-	-	-	3	3	2	L5 L6

SYLLABUS**UNIT I:****15 Hours**

Introduction: Characteristics of Database Approach – Advantages of Using DBMS Approach – Data Models, Schemas, and Instances – Three Schema Architecture – Database Languages and Interfaces – The Database System Environment- Centralized and Client-Server Architectures.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, specialization, generalization using ER Diagrams

Self-Learning Topics: Sub classes, super class and inheritance **CO's–CO1**

UNIT II: **13 Hours**

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. **CO's–CO2**

Self-Learning Topics: Structure of a Relational Database

UNIT III: **15 Hours**

Structured Query Language: Introduction, History of SQL Standard – Commands in SQL – Data Types in SQL – Data Definition Language – Selection Operation, Projection Operation – Aggregate Functions – Data Manipulation Language – Table Modification Commands – Table Truncation – Imposition of Constraints – Join Operation – Set Operations – Views – Sub Query– Embedded SQL. **CO's–CO3**

Self-Learning Topics: Advanced Grouping and Aggregation (GROUP BY with ROLLUP, CUBE, and GROUPING SETS, Filtering groups using HAVING)

UNIT IV: **14 Hours**

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce Codd-normal form (BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF).

Self-Learning Topics: Decomposition Deep Dive (Comparing Lossless vs. Lossy Decomposition, Dependency-Preserving Decomposition: When to compromise) **CO's–CO4**

UNIT V: **13 Hours**

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Testing for Serializability, lock based, time stamp-based protocols, Deadlocks.

Recovery and Atomicity: Log – Based Recovery – Recovery with Concurrent Transactions – Check Points - Buffer Management – Failure with loss of nonvolatile storage. **CO's–CO5**

Self-Learning Topics: Introduction to Indexing Techniques: B+ Trees, operations on B+ Trees, Hash Based Indexing

Board of Studies: Computer Science and Engineering

Approved in BOS No: 03

Approved in ACM No: 03

Text Books:

1. **Database System Concepts**, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill.
2. **Database Management Systems**, Raghu Ramakrishnan, Johannes Gehrke, Tata McGraw Hill, 3rd Edition.

Reference Books:

1. **Database Systems Design**, Implementation and Management, Peter Rob & Carlos Coronel, Cengage Learning, 7th Edition.
2. **Fundamentals of Database Systems**, Ramez Elmasri, Shamkant B. Navathe, Pearson Education.
3. **Database Systems: The Complete Book**, Hector Garcia-Molina, Jeffrey D. Ullman,

Jennifer Widom, Pearson.

4. **An Introduction to Database Systems**, C.J. Date, A. Kannan, S. Swaminathan, Pearson, 8th Edition.

Web References:

1. <http://www.c4learn.com/>
2. <http://www.geeksforgeeks.org/c/>
3. <http://nptel.ac.in/courses/122104019/>
4. <http://www.learn-c.org/>
5. <https://www.tutorialspoint.com/cprogramming/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1 (%)	Internal Assessment # 2(%)
L1	35%	--
L2	40%	--
L3	25%	25%
L4	--	35%
L5	--	25%
L6		15%
Total(%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Database System.
2. What is DBMS?
3. List the advantages of DBMS.
4. Define schema and instance.
5. What is an entity?
6. Define attribute in DBMS.
7. What is a relation?
8. List ACID properties.
9. What is normalization?
10. Define primary key.
11. What is a transaction?
12. What is indexing in DBMS?

L2: Understand

1. Explain database system architecture.
2. Describe the difference between file system and DBMS.
3. Explain ER model with components.
4. Describe data independence.
5. Explain relational model with example.
6. Discuss types of database users.
7. Explain constraints in DBMS.
8. Describe different types of joins.

9. Explain ACID properties of transactions.
10. Discuss functional dependency.
11. Explain normalization concept.
12. Describe deadlock in DBMS.

L3: Apply

1. Draw ER diagram for a student database.
2. Convert ER diagram to relational schema.
3. Write SQL queries for table creation.
4. Perform insert, update, and delete operations in SQL.
5. Apply relational algebra operations on given tables.
6. Write SQL queries using joins. Create a table named EMPLOYEE with appropriate data types and constraints.
7. Implement grouping and aggregation in SQL.
8. Normalize a given table to 3NF.
9. Identify functional dependencies in a relation.
10. Demonstrate transaction scheduling.
11. Apply B+ tree operations on sample data.
12. Write nested SQL queries for a problem.

L4: Analyzing

1. Analyze differences between relational algebra and SQL.
2. Compare 1NF, 2NF, and 3NF.
3. Analyze redundancy in database design.
4. Differentiate centralized and client-server architecture.
5. Compare different types of joins.
6. Analyze functional dependencies in a relation.
7. Examine concurrency control problems.
8. Analyze deadlock conditions.
9. Compare lock-based and timestamp-based protocols.
10. Analyze normalization vs denormalization.
11. Compare B+ tree and hashing techniques.
12. Analyze query optimization techniques.

L5: Evaluating

1. Evaluate the importance of normalization.
2. Justify the need for ER modelling.
3. Evaluate different indexing techniques.
4. Assess transaction management mechanisms.
5. Evaluate concurrency control protocols.
6. Compare different database architectures.
7. Evaluate SQL vs relational algebra.
8. Assess efficiency of query processing methods.
9. Evaluate recovery techniques in DBMS.
10. Judge effectiveness of different join methods.
11. Evaluate trade-offs between normalization and performance.
12. Critically analyze database security issues.

L6: Creating

1. Develop transaction model for ATM system.
2. Design a university database with constraints.
3. Create indexing strategy for large database.
4. Design a scheduling system using DBMS concepts.
5. Develop relational schema from given requirements.
6. Design an e-commerce database with relationships.
7. Create a recovery system for transaction failure scenario.
8. Design a complete database solution for social media platform.

**Chairperson
Board of Studies (CSE)**

R24CSOE03**AGILE METHODOLOGIES****3 0 0 3****(Open Elective-I)****Course Objectives:**

1. Understand the fundamentals of Agile methodology, Agile values, and the Agile Manifesto for effective software development.
2. Apply Agile principles to improve project delivery, customer collaboration, communication, and continuous improvement.
3. Understand Scrum framework, self-organizing teams, sprint planning, and Scrum practices for effective project management.
4. Analyze Extreme Programming (XP) principles and practices for embracing change, improving software quality, and incremental design.
5. Understand Lean, Kanban, and Agile coaching principles for eliminating waste, managing workflow, and continuous process improvement.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE03.1	Explain fundamentals of Agile methodology.	3	2	---	---	1	---	---	---	---	2	---	2	1	L1, L2	
R24CSOE03.2	Explain agile principles.	3	2	1	---	1	---	---	---	1	2	---	2	1	L2, L3	
R24CSOE03.3	Apply Scrum principles	3	3	3	1	2	---	---	---	2	3	1	3	2	L3, L4	
R24CSOE03.4	Apply practices of XP and Incremental design	3	3	3	2	3	---	---	---	2	3	1	3	3	L4, L5	
R24CSOE03.5	Develop methods to eliminate waste	3	3	2	2	3	1	1	---	2	3	2	3	3	L5, L6	

SYLLABUS**UNIT I:****13 Hours**

LEARNING AGILE: Getting Agile into Your Brain, Understanding Agile Values, No Silver Bullet, Agile to the Rescue, Adding Agile Makes a Difference. A Fractured Perspective, How a Fractured Perspective Causes Project Problems. The Agile Manifesto, Purpose behind Each Practice. Individuals and Interactions Over Processes and Tools, Working Software Over Comprehensive Documentation, Customer Collaboration Over Contract Negotiation, Responding to Change Over Following a Plan, Principles Over Practices. Understanding the Elephant, Methodologies Help You Get It All in Place at Once, Where to Start with a New Methodology

Self-Learning Topics: Agile Software Lifecycle, Agile in Startup Companies.**CO's-CO1****UNIT II:****15 Hours****THE AGILE PRINCIPLES:** The 12 Principles of Agile Software, The Customer Is Always

Right...Right? , “Do As I Say, Not As I Said”. Delivering the Project, Better Project Delivery for the Ebook Reader Project. Communicating and Working Together, Better Communication for the Ebook Reader Project. Project Execution—Moving the Project Along, A Better Working Environment for the E-book Reader Project Team. Constantly Improving the Project and the Team. The Agile Project: Bringing All the Principles Together **CO’s–CO2**

Self-Learning Topics: Agile Requirement Engineering, Agile Project Monitoring Tools.

UNIT III:

14 Hours

SCRUM AND SELF-ORGANIZING TEAMS: The Rules of Scrum, Act I: I Can Haz Scrum?, Everyone on a Scrum Team Owns the Project, The Scrum Master Guides the Team’s Decisions, The Product Owner Helps the Team Understand the Value of the Software, Everyone Owns the Project, Scrum Has Its Own Set of Values, Status Updates Are for Social Networks!, The Whole Team Uses the Daily Scrum, Feedback and the Visibility – Inspection –Adaptation Cycle, The Last Responsible Moment, How to Hold an Effective Daily Scrum. Sprinting into a Wall, Sprints, Planning, and Retrospectives, Iterative or Incremental? The Product Owner Makes or Breaks the Sprint, Visibility and Value, How to Plan and Run an Effective Scrum Sprint.

SCRUM PLANNING AND COLLECTIVE COMMITMENT: Not Quite Expecting the Unexpected, User Stories, Velocity, and Generally Accepted Scrum Practices, Make Your Software Useful, User Stories Help Build Features Your Users Will Use, Conditions of Satisfaction, Story Points and Velocity, Burn down Charts, Planning and Running a Sprint Using Stories, Points, Tasks, and a Task Board. Victory Lap, Scrum Values Revisited, Practices Do Work without the Values (Just Don’t Call It Scrum), is Your Company’s Culture Compatible with Scrum Values

Self-Learning Topics: Agile Metrics and Velocity, Scrum Case Studies in IT Industry. **CO’s–CO3**

UNIT IV:

13 Hours

XP AND EMBRACING CHANGE: Going into Overtime, The Primary Practices of XP, Programming Practices, Integration Practices, Planning Practices, Team Practices, Why Teams Resist Changes, and How the Practices Help. The Game Plan Changed, but We’re Still Losing, The XP Values Help the Team Change Their Mindset, XP Helps Developers Learn to Work with Users, Practices Only “Stick” When the Team Truly Believes in Them, An Effective Mindset Starts with the XP Values, The XP Values, Paved with Good Intentions. The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change, The Principles of XP, XP Principles Help You Understand Planning, XP Principles Help You Understand Practices—and Vice Versa, Feedback Loops.

XP, SIMPLICITY, AND INCREMENTAL DESIGN: Code and Design, Code Smells and Anti-patterns (or, How to Tell If You’re Being Too Clever), XP Teams Look for Code Smells and Fix Them, Hooks, Edge Cases, and Code That Does Too Much. Make Code and Design Decisions at the Last Responsible Moment, Fix Technical Debt by Refactoring Mercilessly, and Use Continuous Integration to Find Design Problems, Avoid Monolithic Design, Incremental Design and the Holistic XP Practices. Teams Work Best When They Feel Like They Have Time to Think, Team Members Trust Each Other and Make Decisions Together. The XP Design, Planning, Team, And Holistic Practices for Man Ecosystem Incremental Design Versus Designing for Reuse, When Units Interact in a Simple Way, the System Can Grow Incrementally; Great Design Emerges from Simple Interactions, Final Score. **COs–CO4**

Self-Learning Topics: Agile Coding Standards, Micro services and Agile Development

UNIT V:

13 Hours

LEAN, ELIMINATING WASTE AND SEEING THE WHOLE: Lean Thinking, Commitment, Options Thinking, and Set-Based Development, Creating Heroes and Magical Thinking. Eliminate Waste, use a Value Stream Map to Help See Waste Clearly, gain a Deeper Understanding of the Product, See the Whole, and Find the Root Cause of Problems That You Discover. Deliver As Fast As Possible; Use an Area Chart to Visualize Work in Progress, Control Bottlenecks by Limiting Work in Progress.

KANBAN, FLOW, AND CONSTANTLY IMPROVING: The Principles of Kanban find a Starting Point and Evolve Experimentally from There. Stories Go into the System; Code Comes Out, Improving Your Process with Kanban, Visualize the Workflow, Limit Work in Progress. Measure and Manage Flow, Managing Flow with WIP Limits Naturally Creates Slack. Make Process Policies Explicit So Everyone Is on the Same Page. Emergent Behavior with Kanban.

THE AGILE COACH: Coaches Understand Why People Don't Always Want to Change. The Principles of Coaching. **CO's-CO5**

Self-Learning Topics: Agile Process Improvement Techniques, Agile Coaching Skills

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Agile and Scrum Practices in Software Industries

This expert talk focuses on the practical implementation of Agile and Scrum methodologies in software industries. The session covers Scrum roles, sprint planning, daily stand-up meetings, backlog management, team collaboration, and real-time project execution practices followed in IT companies.

2. Lean, Kanban, and Agile Transformation in IT Projects

This session explains how Lean principles and Kanban techniques are used in modern IT organizations to improve workflow efficiency, eliminate waste, manage work-in-progress, and achieve continuous improvement during Agile transformation processes.

Text Books:

1. Andrew Stellman, Jill Alison Hart, *Learning Agile*, O'Reilly, 2015

Reference Books:

1. Rubin K, *Essential Scrum: A practical guide to the most popular Agile process*, Addison-Wesley, 2013

Web References:

1. NPTEL (SWAYAM): <https://nptel.ac.in>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Agile methodology. (Unit I)
2. What is the Agile Manifesto? (Unit I)
3. List the 12 Agile principles. (Unit II)
4. Define Scrum. (Unit III)
5. What is a Sprint in Scrum? (Unit III)
6. Define Extreme Programming (XP). (Unit IV)
7. What is refactoring? (Unit IV)
8. Define Lean software development. (Unit V)
9. What is Kanban? (Unit V)
10. Define Work in Progress (WIP). (Unit V)

L2: Understand

1. Explain Agile values and their importance. (Unit I)
2. Describe the purpose of the Agile Manifesto. (Unit I)
3. Explain the 12 principles of Agile software development. (Unit II)
4. Describe the roles of Scrum Master and Product Owner. (Unit III)
5. Explain the concept of Daily Scrum. (Unit III)
6. Describe XP values and principles. (Unit IV)
7. Explain code smells and anti-patterns. (Unit IV)
8. Describe Lean principles in software development. (Unit V)
9. Explain Kanban workflow visualization. (Unit V)
10. Describe the role of an Agile Coach. (Unit V)

L3: Apply

1. Apply Agile principles to improve software project communication. (Unit II)
2. Demonstrate how Scrum can be used to manage a software project. (Unit III)
3. Apply user stories and story points in Sprint planning. (Unit III)
4. Use refactoring techniques to improve code quality. (Unit IV)
5. Apply continuous integration in an XP environment. (Unit IV)
6. Use Kanban boards to manage workflow in a project team. (Unit V)

L4: Analysing

1. Analyze the differences between traditional and Agile methodologies. (Unit I)
2. Compare Agile Manifesto values with traditional software practices. (Unit I)
3. Analyze how Agile principles improve project delivery. (Unit II)
4. Compare Scrum and XP methodologies. (Unit III & IV)
5. Analyze the impact of velocity and burn-down charts in Scrum projects. (Unit III)
6. Compare Lean and Kanban approaches. (Unit V)

L5: Evaluating

1. Evaluate the effectiveness of Agile methodologies in software development. (Unit I)
2. Assess the importance of customer collaboration in Agile projects. (Unit II)
3. Evaluate the role of Scrum Master in team success. (Unit III)

4. Assess the effectiveness of XP practices in handling changing requirements. (Unit IV)
5. Evaluate the impact of refactoring on software maintainability. (Unit IV)
6. Assess the advantages of Kanban in workflow management. (Unit V)

L6: Create

1. Design an Agile workflow for a software development project. (Unit I)
2. Develop a Scrum plan for an e-commerce application. (Unit III)
3. Create user stories and Sprint backlog for a mobile app project. (Unit III)
4. Design an XP-based development strategy for a software team. (Unit IV)
5. Develop a Kanban board for project workflow management. (Unit V)
6. Create a Lean-based process improvement plan for software development. (Unit V)

**Chairperson
Board of Studies (CSE)**

R24CSOE04**OPERATING SYSTEMS****3 0 0 3**

(Open Elective-I)

Course Objectives:

1. To understand operating system overview, functions and services.
2. To analyze process concept, inter process communication and synchronization.
3. Illustrate different conditions for deadlock and their possible solutions.
4. To understand secondary storage management.
5. To understand protection and security.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE 04.1	Describe operating system overview, functions, and services	3	2	1	1	2	-	-	-	1	-	2	2	1	L1, L2	
R24CSOE 04.2	Analyze process concept, inter-process communication, and synchronization	3	3	2	1	3	-	-	1	1	-	2	3	2	L2, L3	
R24CSOE 04.3	Analyze process synchronization mechanisms and dead lock handling techniques.	3	3	2	2	2	-	-	1	1	-	2	3	2	L3, L4	
R24CSOE 04.4	Describe methods for mapping secondary storage management.	3	2	2	1	3	-	-	-	1	-	2	3	2	L4, L5	
R24CSOE 04.5	Evaluate protection mechanism and security	3	3	2	1	3	2	2	-	1	-	3	3	3	L5, L6	

SYLLABUS**UNIT I:****14 Hours**

Introduction to Operating Systems, OS Functions and Services, OS Operations and Types, Computing Environments, Open-Source vs Proprietary OS, System Calls, Types of System Calls, OS Design and Structure, OS Booting Process

CO's-CO1

Self-Learning Topics: Evaluation of operating systems, Comparison of Linux, Windows and Android

UNIT II:**13 Hours**

Process Concept, Process States, Process Control Block, Process Scheduling, Context Switching, Inter-Process Communication, Threads, Multithreading Models, Thread Libraries, CPU Scheduling Algorithms, Scheduling Evaluation

CO's-CO2

Self-Learning Topics: Multicourse processing, Real-time scheduling

UNIT III:**13 Hours**

Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classical Synchronization Problems, Deadlock Concept, System Model, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance

CO's-CO3

Self-Learning Topics: Hardware support for synchronization, Spinlocks and barriers, Resource allocation graphs, Case studies of deadlocks

UNIT IV:**14 Hours**

Contiguous Memory Allocation, Paging, Page Tables, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Thrashing, File Concepts, Access Methods, Directory Structures, File Allocation Methods

CO's-CO4

Self-Learning Topics: Cache memory and hierarchy, advanced page replacement algorithms

UNIT V:**14 Hours**

Disk Structure, Disk Scheduling, RAID, Protection Goals, Protection Principles, Access Matrix Model, Security Threats, Network Security Concepts

CO's-CO5

Self-Learning Topics: SSD vs HDD performance, Basics of cryptography in security

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)**CO's/ PO's / PSO's**

1. Operating Systems in Cloud Computing (Virtualization Techniques in Industry Cloud Platforms (AWS, Azure, GCP), Containerization (Docker, Kubernetes) vs Virtual Machines, Operating System Abstractions in Cloud Architecture) CO1–CO5, PO1, PO2, PO3, PO11
2. Real-Time Operating Systems (RTOS) in Industry Applications (RTOS in Automotive Systems (ADAS, AUTOSAR), RTOS for Industrial Automation and Robotics vs General Purpose OS in Embedded Systems) CO1–CO5, PO1, PO2, PO3, PO11
3. Operating Systems in Cybersecurity (OS Hardening Techniques in Enterprises, Secure Boot and Trusted Execution Environments, Role of OS in Endpoint Security Solutions) CO1–CO5, PO1, PO2, PO3, PO11

Text Books:

1. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 10th Edition, John Wiley
2. Operating Systems- A Concept based Approach, D. M. Dhamdhare, 2nd Edition, Tata McGraw-Hill
3. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley

Reference Books:

1. Operating Systems-Internal and Design Principles, Stallings, 5th Edition, Pearson Education/PHI, 2005
2. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
3. UNIX: Concepts and Applications, Sumit Abha Das, 4th Edition, Tata McGraw-Hill
4. Operating System: A Design-Oriented Approach, Charles Patrick Crowley

Web References:

1. <https://archive.nptel.ac.in/courses/106/102/106102132/>
2. https://edurev.in/courses/23057_Operating-System

3. https://www.tutorialspoint.com/operating_system

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1 (Remember)

1. Define Operating System
2. List OS functions and services
3. What is a system call
4. Define process and PCB
5. What is a semaphore
6. Define deadlock
7. What is paging
8. Define virtual memory
9. What is RAID
10. Define protection

L2 (Understand)

1. Explain OS operations and types
2. Describe computing environments
3. Explain process states and lifecycle
4. Describe context switching
5. Explain IPC methods
6. Explain Peterson's solution
7. Describe page replacement algorithms
8. Explain file access methods
9. Describe disk scheduling algorithms
10. Explain access matrix model

L3 (Apply)

1. Apply FCFS scheduling for given processes
2. Demonstrate Round Robin scheduling
3. Apply semaphore to solve producer-consumer problem
4. Solve a deadlock avoidance example
5. Apply paging for address translation
6. Solve FIFO page replacement problem

7. Apply disk scheduling algorithms
8. Demonstrate access control using access matrix

L4 (Analyze)

1. Compare monolithic and microkernel OS structures
2. Analyze system call types
3. Compare process vs thread performance
4. Analyze CPU scheduling algorithms
5. Compare synchronization techniques
6. Analyze deadlock conditions and system model
7. Compare paging and segmentation
8. Analyze thrashing
9. Analyze disk scheduling performance
10. Compare protection mechanisms and security threats

L5 (Evaluate)

1. Evaluate open-source vs proprietary OS
2. Justify OS structure selection
3. Evaluate CPU scheduling algorithms
4. Justify use of multithreading
5. Evaluate deadlock prevention vs avoidance
6. Access memory management techniques
7. Justify page replacement algorithms
8. Evaluate file allocation methods
9. Assess RAID levels
10. Evaluate protection and security mechanisms

L6 (Create)

1. Design an operating system architecture
2. Propose improvements to OS services
3. Design a scheduling algorithm
4. Develop a multithreaded application model
5. Design synchronization solution
6. Create a deadlock-free system
7. Design memory management system
8. Develop file system structure
9. Propose disk scheduling algorithm
10. Design a secure system

Chairperson
Board of Studies (CSE)

Course Objectives:

1. Understand the concepts of scripting languages for developing Web Scripting.
2. Create database connections using PHP and build the website for the world.
3. Analyse the internet ware application, security issues and frame work for application.
4. Database driven Web Applications using PHP.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE05.1	Identify the basic programming constructs of PHP.	3	2	2	2	2	-	-	-	-	-	-	3	2	L1, L2	
R24CSOE05.2	Develop programs using functions, strings and arrays.	3	3	2	2	2	-	-	-	-	-	-	3	2	L2, L3	
R24CSOE05.3	Apply object-oriented principles in PHP	3	3	2	2	2	-	-	-	-	-	-	3	3	L3, L4	
R24CSOE05.4	Design interactive web pages by using JQuery & AJAX	3	2	2	2	3	-	-	-	-	-	-	3	2	L4, L5	
R24CSOE05.5	Design data driven applications by using PHP	3	2	3	2	3	-	-	-	-	-	-	3	2	L5, L6	

SYLLABUS**UNIT-I:****13 Hours**

Introduction to PHP: History, Web Browser, Web Server – XAMPP, Installation and Configuration files. Evaluation of PHP, Basic syntax, defining variable and constant, PHP Data types, Operators and Expressions. Decisions and loops: Making Decisions, doing Repetitive task with looping, mixing decisions and looping with HTML.

COs–CO1

Self-Learning Topics: Advanced XAMPP Configuration

UNIT- II: FUNCTIONS, STRING & ARRAY**15 Hours**

Functions: What is a function, Define a function, Call by value and Call by reference and Recursive functions? **String:** Creating and accessing String, Searching & Replacing String, Formatting String, String Related Library functions **Array:** Anatomy of an Array, Creating Index based and Associative array, Accessing array element, Looping with Index based array, Looping with associative array using each() and foreach(), some useful library functions.

COs–CO2

Self-Learning Topics: Building Reusable Utilities

UNIT –III:**15 Hours**

ADVANCE PHP: Introduction: Objects, declaring a class, the new keyword and constructor,

Destructor, Access method and properties using this variable. Public, private, protected properties and methods, Static properties and method, Class constant, Inheritance & code, reusability, Polymorphism, Parent::& self:: keyword, Instance of operator, Abstract method and class, Interface, Final. Exception Handling: Understanding Exception and error. Try, catch, and throw. **COs-CO3**

Self-Learning Topics: Evaluate OOP vs. procedural programming

UNIT-IV: PHP SCRIPT

12 Hours

JQuery: Introduction to JQuery, Validation using JQuery, JQuery Forms, JQuery Examples.

AJAX: Introduction to AJAX, PHP with AJAX, Working with database.

COs-CO4

Self-Learning Topics: Evaluate AJAX vs traditional page reloads

UNIT-V: PHP WEB SERVICES

15 Hours

Handling HTML form with PHP: Capturing Form Data, Dealing with Multi-value file, generating File uploaded form, redirecting a form after submission. Web Features: Sessions, Forms GET and POST data, Cookies, HTTP Headers. Database Connectivity with

MySQL: Introduction to RDBMS, Connection with MySQL Database, Performing basic database operations(DML) (Insert,Delete,Update,Select),Setting query parameter, Executing query.

Self-Learning Topics: Perform **CRUD** operations using **PHP & My SQL**

COs-CO5

Board of Studies: Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

COs

POs / PSOs

1. Evaluate session vs. cookies for user management

CO5

PO1, PO2, PO3

Text Books:

1. Steven Holzner, —PHP: The Complete Reference, McGraw-HillEducation, 2007.
2. Kevin Tatroe, Peter Mac Intyre, Rasmus Lerdorf, —Programming in PHP, O'Reilly, 3rd Edition, 2013.
3. Luke Welling, Laura Thomson, —PHP and My SQL Web Development, Pearson Education, 5th Edition.

Reference Books:

1. HTML5BlackBook, CoversCSS3, Java Script, XML, XHTML, AJAX, PHP and jQuery DT Editorial Services Dream Tech Publications.
2. Lynn Beighley, Michael Morrison, —Head First PHP &My SQL: A Brain-Friendly Guidel, O'Reilly, 1st Edition.
3. David Powers, —PHP Solutions: Dynamic Web Design Made Easy, Après, 3rd Edition.

Web References:

1. https://www.w3schools.com/php/php_intro.asp
2. https://www.tutorialspoint.com/php/php_introduction.htm
3. <https://www.tutorialrepublic.com/php-tutorial/php-variables.php>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--

L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define PHP.
2. What is a web browser?
3. List configuration files in PHP.
4. List PHP data types
5. What is a function?
6. What is jQuery?
7. What is AJAX?
8. What is session?

L2: Understand

1. What is XAMPP?
2. What are variables and constants?
3. Define operators and expressions.
4. What is decision making in PHP?
5. Define recursive function.
6. Types of arrays in PHP.
7. What is destructor?

L3: Apply

1. Describe installation and configuration of XAMPP server.
2. Explain PHP syntax, variables, constants, and data types with examples.
3. Explain decision-making and looping constructs with examples and HTML integration.
4. Discuss call by value and reference with suitable examples.
5. Explain array traversal using loops.
6. Explain OOP concepts in PHP with examples.
7. Explain static properties, methods, and constants.
8. Demonstrate PHP with AJAX for database operations.
9. Explain handling HTML forms in PHP.
10. Explain GET and POST methods with use cases.
11. Explain CRUD operations with PHP and My SQL.

L4: Analysing

1. Discuss operators and expressions in PHP.
2. Compare different looping statements in PHP.
3. Discuss call by value and reference with suitable examples.
4. Compare indexed and associative arrays.
5. Explain inheritance and code reusability.
6. Discuss polymorphism and interfaces.

7. Difference between GET and POST.
8. Discuss sessions and cookies with examples.

L5: Evaluating

1. Evaluate the advantages and limitations of PHP compared to other server-side scripting languages.
2. Assess the effectiveness of using XAMPP as a local web server environment for PHP development.
3. Critically evaluate the role of web browsers and web servers in executing PHP applications.
4. Assess the usefulness of recursive functions in PHP.
5. Critically analyze the role of arrays and functions in simplifying complex string operations.
6. Critically evaluate the role of classes and objects in building scalable PHP applications.
7. Assess the use of parent:: and self:: keywords in PHP.
8. Assess the difference between errors and exceptions in PHP.

L6: Create

1. Write a program using recursion?
2. Write a PHP program demonstrating inheritance and polymorphism.
3. Develop a small application using AJAX and PHP.
4. Design a form handling system with validation and database storage.

Chairperson
Board of Studies (CSE)

R24CSPC18**COMPUTER NETWORKS LAB****0 0 3 1.5**

(Computer Science and Engineering)

Course Objectives:

1. Understand and implement basic networking protocols and algorithms.
2. Develop network programs using C (client-server, routing).
3. Simulate and analyze network performance and behavior.
4. Use tools to capture and study network traffic.

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs													
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	Do K
R24CS PC18.1	Develop and implement computer networking concepts such as framing methods, error detection techniques, routing algorithms, socket programming, and sliding window protocols using C programming.	3	3	3	1	2	0	0	0	1	0	0	3	2	L1, L2
R24CS PC18.2	Analyze and simulate network routing protocols, broadcast mechanisms, & networking algorithms using C programming and use simulation tools such as NS2/NS3, Packet Tracer, or equivalent Simulators to find the network performance parameters	3	3	2	3	2	0	0	0	1	1	0	3	3	L2, L3
R24CS PC18.3	Capture, monitor, filter, and troubleshoot network traffic using networking tools such as Wireshark, raw sockets, ARP, and DNS client-server applications to understand network communication and diagnostics.	2	3	1	2	2	0	0	0	1	3	1	3	3	L3

Board of Studies: Computer Science and Engineering

Approved in BOS No: 03

Approved in ACM No: 03

Developing the following programs:

- | | |
|---|------------------|
| Week 1: | CO's:CO1 |
| 1. Write a C Program to implement the data link layer framing methods such as:
i) Character stuffing ii) Bit stuffing | |
| Week 2: | CO's:CO1 |
| 2. Write a C Program to implement on a data set of characters the three CRC polynomials – CRC-12, CRC-16and CRC CCIP. | |
| Week 3: | CO's:CO1 |
| 3. Write a C Program to implement Dijkstra 's algorithm to compute the shortest path through a graph. | |
| Week 4: | CO's:CO1 |
| 4. Write a C Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes). | |
| Week 5: | CO's:CO1 |
| 5. Write a C Program of client-server application for chat using UDP or TCP | |
| Week 6: | CO's:CO1 |
| 6. Write a C program to perform sliding window protocol for
i) Go Back - N ii) Selective Repeat | |
| Week 7: | CO's: CO2 |
| 7. Write a C Program to Simulate the Open Shortest Path First routing protocol based on the cost assigned to the path | |
| Week 8: | CO's: CO2 |
| 8. Write a C Program to implement Broadcast tree by taking subnet of hosts. | |
| Week 9: | CO's: CO2 |
| 9. Write a C Program to Implementing Routing Protocols using border gateway protocol (BGP) | |
| Week 10: | CO's: CO2 |
| 10. Write a C Program to Get the MAC or Physical address of the system using Address Resolution Protocol. | |
| Week 11: | CO's: CO3 |
| 11. Write a C program to develop a DNS client server to resolve the given hostname | |
| Week 12: | CO's: CO3 |
| 12. Write a C Program to using raw sockets to implement packet capturing and filtering | |
| Week 13: | CO's:CO3 |
| 13. Install Wireshark Tool on PC and
a) Capture network traffic | |

- b) examine the traffic captured by Wire shark

Week 14:

CO's:CO3

14. Install Wireshark Tool on PC and
 - a) Filter and examine only ICMP traffic using Wireshark Tool
 - b) Determine default gateway address of your network using Wireshark Tool

Week 15:

CO's:CO3

15. Simulate a three nodes' point-to-point network with duplex links between them. Set the queue size varies the bandwidth and find the number of packets dropped
ns2/ns3/CISCO Packet Tracet /OPNET/any other network simulator may be used for simulation experiments.

Reference Books:

1. W. Richard Stevens – *UNIX Network Programming*, 2nd Edition, 1998
2. Donahoo & Calvert – *TCP/IP Sockets in C*, 2nd Edition, 2009
3. Behrouz A. Forouzan – *TCP/IP Protocol Suite*, 4th Edition, 2010
4. William Stallings – *Data and Computer Communications*, 10th Edition, 2013
5. Yashavant Kanetkar – *Let Us C*, 16th Edition, 2020 (varies by print)
6. Douglas Comer – *Internetworking with TCP/IP*, Vol. 1, 6th Edition, 2014

Online Learning Resources:

1. <https://www.geeksforgeeks.org/computer-network-tutorials/>
2. https://www.tutorialspoint.com/data_communication_computer_network/index.htm
3. <https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-computer-networking.html>
4. <https://nptel.ac.in/courses/106105081> (Computer Networks – NPTEL)
5. <https://www.javatpoint.com/computer-network-tutorial>
6. <https://www.wireshark.org/docs/> (Wireshark documentation)

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. Develop strong **problem-solving and logical thinking skills** to tackle algorithmic challenges efficiently.
2. Improve ability to design **optimized solutions** within given time and space constraints.
3. Enhance coding speed, accuracy, and debugging skills under **competitive and time-bound environments**.
4. Apply key algorithmic paradigms such as **greedy, divide and conquer, dynamic programming, backtracking, and graph algorithms** to solve problems.
5. Analyze problem statements, identify constraints, and select the **most efficient algorithmic approach**.
6. Strengthen understanding of **bit manipulation, string processing, and mathematical techniques** used in programming contests.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSPC 19.1	Identify type of the problem and Apply divide-and-conquer, searching and number-theoretic algorithms to solve problems	3	3	2	1	2	1	-	-	1	-	2	-	-	L1, L2	
R24CSPC 19.2	Design and implement solutions using dynamic programming, pattern matching, graph traversal, and greedy techniques	3	3	3	2	3	1	-	1	1	1	2	-	-	L2, L3	
R24CSPC 19.3	Develop solutions using backtracking and combinatorial optimization techniques for complex constraint-based problems	3	3	3	2	2	1	-	2	1	1	2	-	-	L3	

Developing the following programs:**Week 1:****CO's:CO1**

1. Given an array of integers (which may include negative numbers), find the maximum sum of a contiguous sub array using the Divide and Conquer approach.

Week 2:**CO's-CO1**

The Median of two sorted arrays

Program to find the median of two sorted arrays of same size and different size are discussed here. Firstly, let us see what is median of the array? Median is an element which divides the array into two parts - left and right. So, the number of elements on the left side of the array will be equal or less than the number of elements on the right side. Now, let us consider the case of an array with

odd number of elements. Array = [9,11,16,7,2] Sorted array = [2,7,9,11,16]. In this case, the median of this array is 9, since it divides the array into two parts: [2,7] and [11,16]. Further, let us consider the case of an array with even elements. Array = [1,2,3,4,5,6]. In such a case, we will take the average between the last element of the left part and the first element of the right part. In this case, the median equals $= (3 + 4) / 2 = 3.5$.

Input Format: The input should contain 3 lines.

First line of the input should contain two integer values which specify the number of elements in array1 and array2.

Second line of the input should contain the elements of the first array.

Third line of the input should contain the elements of the second array.

Output Format:

The output should print only the median value

Week 3:

CO's-CO1

We can modify the binary search algorithm to perform division of two numbers by defining the range [0, infinity], which serves as initial low and high for binary search. The goal is to find a value mid such that: $x / y = \text{mid}$ or $x \approx y * \text{mid}$ Based on comparison between x and y * mid: If y * mid is approximately equal to x, return mid

1. If $y * \text{mid} < x$, update low = mid
2. If $y * \text{mid} > x$, update high = mid

Sample Input: 22 7

Sample Output: 3.14286

Week 4:

CO's-CO1

Consider the following algorithm to generate a sequence of numbers:

Start with an integer n. If n is even, divide it by 2. If n is odd, multiply it by 3 and add 1. Repeat this process with the new value of n, terminating when $n = 1$.

For example, for $n = 22$, the sequence is: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that this algorithm will terminate at $n = 1$ for every integer n. For an input n, the cycle length of n is the number of values generated including 1.

Given two integers i and j, determine the maximum cycle length for all numbers between i and j (inclusive).

Sample Input:

```
1    10
100  200
201  210
900   1000
```

Sample Output:

```
1  10 20
100 200 125
201 210 89
900 1000 174
```

Week 5:

CO's-CO1

Euclid Problem

From Euclid's theorem, it is known that for any positive integers A and B, there exist integers X and Y such that:

$$AX + BY = D$$

Where D is the greatest common divisor (GCD) of A and B.

The problem is to find the corresponding values of X, Y, and D for given A and B.

If multiple solutions exist, output the pair (X, Y) such that:

1. $X \leq Y$
2. $|X| + |Y|$ is minimized

Input Format: Each line contains two integers A and B separated by a space.

Constraints: $A, B < 1,000,000,001$

Output Format: For each input line, print three integers X, Y, and D separated by a space.

Sample Input:

4 6
17 17

Sample Output:

-1 1 2
0 1 17

Week 6:

CO's-CO2

Matching String with Wildcard Pattern: Check whether a given input string matches a pattern containing wildcard characters. The pattern may include:

1. $?$ → matches exactly one character
2. $*$ → matches any sequence of characters (including zero characters)

Sample Input:

abcabcccd
?c*d

Sample Output: 1

Week 7:

CO's-CO2

Determine the **minimum time required to rot all fresh oranges** in the grid.

If it is impossible to rot all oranges, return -1.

Given a grid of dimension $n \times m$, each cell in the grid can have one of the following values:

1. 0 → Empty cell
2. 1 → Cell contains a fresh orange
3. 2 → Cell contains a rotten orange

A rotten orange can rot adjacent fresh oranges in **one unit of time**. The infection spreads in four directions:

1. Up → $[i-1, j]$
2. Down → $[i+1, j]$
3. Left → $[i, j-1]$
4. Right → $[i, j+1]$

Input Format: The input consists of a 2D grid of integers of size $n \times m$

Each element is 0, 1, or 2

Output Format: Print a single integer representing the minimum time required

Print -1 if all oranges cannot be rotted

Sample Input:

Grid = $\{\{0, 1, 2\},$
 $\{0, 1, 2\},$
 $\{2, 1, 1\}\}$

Sample Output: 1

Week 8:

CO's-CO2

Coin Problem

Given a value **V** (in cents), and a set of **N coin denominations** $C = \{C_1, C_2, \dots, C_N\}$, you have an **infinite supply** of each type of coin.

Your task is to:

1. Find the minimum number of coins required to make the value **V**
2. Print the coins used in one optimal solution. If it is not possible to make the value, print -1.

Input Format: The first line contains an integer **T**, the number of test cases

For each test case:

1. First line contains two integers **V** and **N**
2. Second line contains **N integers**, representing coin denominations $C[i]$

Output Format: For each test case:

1. Print the coins used in an optimal solution (space-separated)
2. In the next line, print the **minimum number of coins** required
3. If it is not possible to form the value, print -1

Sample Input:

```
1
7 2
2 1
```

Sample Output:

```
2 2 2 1
4
```

Week 9:

CO's-CO2

Spirally Traversing a Matrix

Given a matrix of size $r \times c$, traverse the matrix in **spiral order**.

Spiral traversal means visiting elements in the following order:

1. Left to right across the top row
2. Top to bottom along the right column
3. Right to left across the bottom row
4. Bottom to top along the left column
5. Repeat the process for the inner sub matrix

Input Format: First line contains two integers' **r** and **c** (number of rows and columns)

Next **r lines** contain **c integers each**, representing the matrix

Output Format: Print the elements of the matrix in **spiral order** (space-separated)

Sample Input:

```
r = 4, c = 4
matrix [][] = {
{1, 2, 3, 4},
{5, 6, 7, 8},
{9, 10, 11, 12},
{13, 14, 15, 16}
}
```

Sample Output: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Week 10:

CO's-CO2

Longest Common Sub sequence: Given two strings, find the length of their **Longest Common Subsequence (LCS)**.

Sample Input:

ABCB DAB

BDCABA

Sample Output: 4

Week 11:

CO's-CO2

Given a connected, undirected weighted graph, find the **Minimum Spanning Tree (MST)** using **Prim's Algorithm**. Also compute the **total weight** of the MST.

Week 12:

CO's-CO2

You are given a weighted graph:

Vertices: A, B, C, D

Edges:

1. A → B (4)
2. A → C (1)
3. C → B (2)
4. B → D (1)
5. C → D (5)

Find the shortest distance from **A to all other vertices** using Dijkstra's algorithm.

Week 13:

CO's-CO3

Tug of war is a contest of brute strength where two teams pull in opposite directions on a rope. The team that succeeds in pulling the rope in their direction is declared the winner.

A tug of war is being arranged for an office picnic. The participants must be divided into two teams such that:

1. Every person is assigned to exactly one team.
2. The number of people in the two teams must not differ by more than one.
3. The total weight of each team should be as equal as possible.

Input Format: The first line contains a single integer **T**, the number of test cases.

Each test case consists of:

1. A blank line
2. A line containing an integer **n** (number of people)
3. **n** lines follow, each containing an integer representing the weight of a person

Output Format: For each test case:

1. Print two integers: the total weights of the two teams.
2. If the weights differ, print the **smaller value first**.
3. Print a blank line between outputs of consecutive test cases.

Sample Input: 1

3

100

90

200

Sample Output:

190 200

Week 14:

CO's-CO3

Problem Statement

Given an integer **N**, find the position of the **first set bit (1)** from the **right side** in its binary representation.

1. The position is **1-based indexing** (rightmost bit is position 1).
2. If there is **no set bit** in the integer (i.e., $N = 0$), return **0**.

Input Format: A single integer **N**

Output Format: Print an integer representing the position of the first set bit from the right. Print 0 if no set bit is present

Sample Input: N=18

Sample Output: 2

Week 15:

CO's-CO3

The N-Queens problem is the problem of placing **N chess queens** on an $N \times N$ **chessboard** such that no two queens attack each other.

Two queens attack each other if they are placed:

1. in the same row
2. in the same column
3. in the same diagonal

Given an integer **N**, determine all possible ways to place **N queens** on the board such that the above conditions are satisfied.

Input Format: **N**

Output Format:

- a) Print all possible solutions.
- b) Represent the board using:
 - i). 'Q' for a queen
 - ii). '.' for an empty space

Print each solution on a new line (or as a matrix format).

Also print the **total number of solutions**.

Sample Input: 4

Sample Output:

Solution 1:

```
. Q . .
. . . Q
Q . . .
. . Q .
```

Solution 2:

```
. . Q .
Q . . .
. . . Q
. Q . .
```

Total Solutions: 2

Board of Studies: Computer Science and Engineering

Approved in BOS No: 03

Approved in ACM No: 03

Reference Books:

1. “Data Structures and Algorithms Made Easy” by Narasimha Karumanchi, Career Monk.
2. “Programming challenges” by Steven S. Skiena Miguel A. Revilla,
3. The Programming Contest Training Manual, Springer.

Online Learning Resources:

1. <https://www.geeksforgeeks.org/data-structures/>
2. <https://www.coursera.org/specializations/data-structures-algorithms>
3. <https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures-and-algorithms-in-java>
4. <https://www.simplilearn.com/free-data-structures-algorithms-course-skillup>
5. <https://www.scholarhat.com/free-course/data-structures-algorithms-course>

**Chairperson
Board of Studies (CSE)**

R24CSSCO3**FULLSTACK DEVELOPMENT-2****0 0 3 1.5**

(Common to CSE, CSE (DS), CSE (AI &ML))

Course Objectives:

The main objectives of the course are to

1. Understand and apply ReactJS concepts to build dynamic user interfaces.
2. Develop interactive frontend applications with state management and API integration.
3. Build backend services and RESTful APIs using Node.js and Express.js.
4. Perform database operations and optimization using MongoDB.
5. Design and develop full-stack MERN applications with authentication and real-time features

Course Outcomes:

Course Code	Course Outcomes	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSSC 03.1	To understand and apply frontend development concepts using ReactJS for building dynamic and interactive user interfaces	2	2	2	1	3	-	-	2	2	1	2	3	2	L1, L2	
R24CSSC 03.2	To develop backend applications and RESTful APIs using Node.js and Express.js with proper request handling and middleware.	3	3	3	2	3	-	-	2	2	2	2	3	3	L2, L3	
R24CSSC 03.3	To design and implement full-stack MERN applications with MongoDB for data management, including authentication and real-time features.	3	3	3	3	3	-	-	3	2	2	2	3	3	L3	

WEEK-1

Introduction to ReactJS

JSX syntax

Functional components

useState Hook

Event handling in React

Component rendering lifecycle basic

1. Write a ReactJS program to create a Counter Application with Increment, Decrement, and Reset functionality

CO's-CO1**WEEK-2**

State management with use State

Handling lists and keys in React

Conditional rendering

Forms and input handling

Basic CRUD operations in frontend

1. Develop a ReactJS application to implement a Todo List with add, delete, and mark-as-completed features **CO's-CO1**

WEEK-3

Controlled components

Form handling in React

Input validation techniques

Regular expressions for email and password

Error handling and displaying messages

1. Write a ReactJS program to create a form handling application with validation (Name, Email, Password) **CO's-CO1**

WEEK-4

Fetch API or Axios

useEffect Hook

Working with APIs GET requests

JSON data handling

Search and filter logic

debouncing optional

1. Write a ReactJS program to fetch and display data from a public API
2. Write a ReactJS program to implement a search filter functionality **CO's-CO1**

WEEK-5

Props in React

Component communication parent to child and child to parent

Lifting state up

Reusable components

Prop drilling and intro to Context API

1. Write a ReactJS program to demonstrate props and state communication between components **CO's-CO1**

WEEK-6

Introduction to Node.js

Core modules http

Creating server using http.createServer

Request and response handling

Basics of backend architecture

1. Write a Node.js program to create a simple HTTP server that returns a response message **CO's-CO2**

WEEK-7

File System module fs

read File write FileappendFile

Synchronous vs asynchronous file handling

Event driven architecture

Event Emitter class

1. Write a Node.js program to perform file system operations (create, write, read, append)
2. Write a Node.js program to implement Event Emitter with custom events **CO's-CO2**

WEEK-8

Callbacks
Promises
Async and Await
Error handling using try catch
Event loop basics

1. Write a Node.js program to demonstrate asynchronous programming using callbacks, promises, or async/await **CO's-CO2**

WEEK-9

Introduction to Express.js
REST API principles
Routing GET POST PUT DELETE
JSON handling
Testing using Postman

1. Develop an Express.js application to create a REST API for Books (CRUD operations) **CO's-CO2**

WEEK-10

Middleware concept in Express
Types of middleware
Route parameters req.params
Query strings req.query
Request lifecycle

1. Write an Express.js program to implement a middleware function for logging requests (method, URL, time) **CO's-CO2**
2. Write an Express.js program to handle route parameters and query strings **CO's-CO2**

WEEK-11

Error handling middleware
HTTP status codes
CORS Cross Origin Resource Sharing
API security basics

1. Write an Express.js program to implement error handling middleware
2. Write an Express.js program to enable and configure CORS in the application **CO's-CO2**

WEEK-12

Introduction to MongoDB
Collections and documents
CRUD operations insert and find
MongoDB shell or Compass
BSON vs JSON

1. Write MongoDB queries to perform CRUD operations on a Students collection **CO's-CO3**
2. Write MongoDB queries to find records based on conditions (age, marks, etc.) **CO's-CO3**

WEEK-13

Update and delete queries
Query operators like gt,lt, in...
Aggregation framework
Indexing and performance optimization

1. Write MongoDB queries to perform update and delete operations on documents **CO's-CO3**
2. Write MongoDB aggregation queries to find:
 - i) Total number of students
 - ii) Average marks**CO's-CO3**

3. Explain and implement indexing in MongoDB for performance optimization CO's-CO3

WEEK-14

MERN stack architecture MongoDB Express React Node

JWT Authentication

REST API integration with frontend

State management Context API or Redux basics

Socket.io for real time apps

MVC structure

1. Develop a full-stack MERN CRUD application for Student Management System
2. Develop a Login and Registration system using JWT authentication
3. Create a Blog application using MERN stack
4. Develop an E-commerce backend API using Node.js, Express, and MongoDB
5. Build a real-time chat application using MERN stack and Socket.io

CO's-CO3

Board of Studies : Computer Science and Engineering

Approved in BoS No :03

Approved in ACM No:03

Reference Books:

- 1.B.M. Harwani – Node.js, Express & MongoDB for Web Development
2. Ethan Brown – Web Development with Node and Express
- 3.Kristina Chodorow – MongoDB: The Definitive Guide
- 4.Alex Banks & Eve Porcello – Learning React
- 5.Mario Casciaro & Luciano Mammino – Node.js Design Patterns

Online Learning Resources:

1. <http://www.w3school.in>
2. ExpressJS - <https://www.tutorialspoint.com/expressjs>
3. ReactJS - <https://www.w3schools.com/REACT> (and) <https://react.dev/learn#>
4. MongoDB - <https://learn.mongodb.com/learning-paths/introduction-to-mongodb>

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To enhance problem-solving skills using efficient algorithms.
2. To develop advanced programming skills in modern languages (Java/Python).
3. To apply algorithmic techniques such as dynamic programming and greedy methods.
4. To improve debugging, testing, and code optimization skills.
5. To encourage participation in competitive programming and coding challenges.
6. To develop the ability to write clean, readable, and maintainable code.
7. To apply coding knowledge to solve real-world problems.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1		PSO2
R24CSPC 20.1	Apply basic programming concepts and problem-solving techniques to develop simple computational solutions using control structures, arrays, and strings.	3	3	2	1	2	1	1	1	1	1	1	-	-	L1, L2
R24CSPC 20.2	Design efficient algorithms using appropriate data structures and analyze their time and space complexity.	2	3	3	2	3	2	1	2	2	2	2	-	-	L2, L3
R24CSPC 20.3	Apply recursion, backtracking, and dynamic programming techniques to solve standard and moderately complex programming problems.	1	3	3	3	3	2	1	2	2	2	3	-	-	L3

SYLLABUS**Developing the following programs:****Week 1:****CO's-CO1**

Given an array of integers of size N, perform a **left rotation by K positions**. In a left rotation, elements are shifted to the left, and the elements at the beginning move to the end. Input Format: First line contains an integer N (size of array) Second line contains N space-separated integers Third line contains an integer K (number of rotations). Output Format: Print the array after left rotation (space-separated).

Week 2:**CO's-CO1**

Airport security officials confiscated several items of the passengers at the security checkpoint. All the items have been dumped into a huge box (array). Each item possesses a certain amount of risk [0,1,2]. Here, the risk severity of the items represents an array [] of N number of integer values. The task here is to sort the items based on their levels of risk in the array. The risk values range from 0 to 2.

Input:

7 -> Value of N

[1, 0, 2, 0, 1, 0, 2]-> Element of arr[0] to arr[N-1], while input each element is separated by new line.

Output:

0 0 0 1 1 2 2 -> Element after sorting based on risk severity

Week 3:

CO's-CO1

N-base notation is a system for writing numbers that uses only n different symbols, This symbols are the first n symbols from the given notation list(Including the symbol for o) Decimal to n base notation are (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:A,11:B and so on up to 35:Z)

Implement the following function: char* DectoNBase (int n, int num): The function accept positive integer n and num Implement the function to calculate the n-base equivalent of num and return the same as a string Steps:

1. Divide the decimal number by n, Treat the division as the integer division
2. Write the remainder (in n-base notation)
3. Divide the quotient again by n, Treat the division as integer division
4. Repeat step 2 and 3 until the quotient is 0
5. The n-base value is the sequence of the remainders from last to first

Assumption:

$1 < n \leq 36$

Input

n: 12

num: 718

Output

4BA

Week 4:

CO's-CO1

You are given **N items**, each having a **weight** and a **value**. You are also given a knapsack with a maximum capacity **W**.

You have to select items such that:

1. The total weight does not exceed **W**
 2. The total value is maximized
 3. Each item can be selected **at most once**
- Find the **maximum value** that can be obtained.

Sample Input:

3 4

1 10

2 15

3 40

Sample Output:50

Week 5:**CO's-CO2**

The Binary number system only uses two digits, 0 and 1 and number system can be called binary string. You are required to implement the following function:

```
int Operations Binary String (char* str);
```

The function accepts a string str as its argument. The string str consists of binary digits separated with an alphabet as follows:

1. – A denotes AND operation
2. – B denotes OR operation
3. – C denotes XOR Operation

You are required to calculate the result of the string str, scanning the string to right taking one operation at a time, and return the same.

Note:

1. No order of priorities of operations is required
2. Length of str is odd
3. If str is NULL or None (in case of Python), return -1

Input:

str: 1C0C1C1A0B1

Output:

1

Explanation:

The alphabets in str when expanded becomes “1 XOR 0 XOR 1 XOR 1 AND 0 OR 1”, result of the expression becomes 1, hence 1 is returned.

Sample Input:

0C1A1B1C1C1B0A0

Output:

0

Week 6:**CO's-CO2**

The function accepts two positive integers ‘r’ and ‘unit’ and a positive integer array ‘arr’ of size ‘n’ as its argument ‘r’ represents the number of rats present in an area, ‘unit’ is the amount of food each rat consumes and each ith element of array ‘arr’ represents the amount of food present in ‘i+1’ house number, where $0 \leq i$

Note:

1. Return -1 if the array is null
2. Return 0 if the total amount of food from all houses is not sufficient for all the rats.
3. Computed values lie within the integer range.

Input:

1. r: 7
2. unit: 2
3. n: 8
4. arr: 2 8 3 5 7 4 1 2

Output:4**Week 7:****CO's-CO2**

def difference of Sum(n, m)The function accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between

sum of integers not divisible by n with sum of numbers divisible by n.

Assumption:

1. $n > 0$ and $m > 0$
2. Sum lies between integral range

Input

n:4

m:20

Output

90

Week 8:

CO's-CO2

Implement the following Function

def Product Smallest Pair (sum, arr)

The function accepts integers sum and an integer array arr of size n. Implement the function to find the pair, (arr[j], arr[k]) where $j \neq k$, such that arr[j] and arr[k] are the least two elements of array (arr[j] + arr[k] $\leq$ sum) and return the product of element of this pair

NOTE

1. Return -1 if array is empty or if $n < 2$
2. Return 0, if no such pairs found
3. All computed values lie within integer range

Sample Input

sum:9

size of Arr = 7

Arr:5 2 4 3 9 7 1

Sample Output

2

Week 9:

CO's-CO2

Given N gold wires, each wire has a length associated with it. At a time, only two adjacent small wires are assembled at the end of a large wire and the cost of forming is the sum of their length. Find the minimum cost when all wires are assembled to form a single wire.

For Example:

Suppose, Arr[]={7,6,8,6,1,1,}

{7,6,8,6,1,1}-{7,6,8,6,2} , cost =2

{7,6,8,6,2}- {7,6,8,8}, cost = 8

{7,6,8,8} – {13,8,8}, cost=13

{13,8,8} -{13,16}, cost=16

{13, 16} – {29}, cost =29

2+8+13+16+29=68

Hence, the minimum cost to assemble all gold wires is 68.

Week 10:

CO's-CO2

Given a **ingly linked list**, determine whether it contains a cycle. A cycle exists if a node in the list can be reached again by continuously following the next pointer. Input Format: First line contains an integer N (number of nodes) Second line contains N space-separated integers (node values) Third line contains an integer POS.POS indicates the position (0-based index) where the last node connects

Week 11:

CO's-CO2

Given an array of integers of size N, find the **maximum sum of any contiguous sub array of size K**. Input Format: First line contains an integer N (size of array) Second line contains N space-separated integers Third line contains an integer K (window size). Output Format: Print a single integer — maximum sum of sub array of size K

Week 12:

CO's-CO2

Given a string S, find the total number of **distinct substrings** of the string.
A substring is a contiguous sequence of characters within the string.

Sample Input: abc

Sample Output: 6

Week 13:

CO's-CO2

You are given a rod of length N and an array **price []**, where price[i] represents the price of a rod piece of length i + 1. You can cut the rod into any number of pieces. Your task is to **maximize the total profit** by choosing the best way to cut the rod.

Sample Input:

4

2 5 7 8

Sample Output:

10

Week 14:

CO's-CO3

Bellman-Ford Algorithm: Given a directed graph with **V vertices** and **E edges**, and a source vertex S, find the **shortest distance from S to all other vertices**. The graph may contain **negative edge weights**. If a **negative weight cycle** exists, report it.

Week 15:

CO's-CO3

Check if string can be segmented into dictionary words.

Sample Input:

"applepenapple"

dictionary = ["apple", "pen"]

Sample Output: True

Board of Studies : Computer Science and Engineering

Approved in BOS No : 03

Approved in ACM No: 03

Reference Books:

1. Ashok N. Kamthane – Programming and Data Structure
2. Reema Thareja – Data Structures Using C
3. Yashavant Kanetkar – Let Us C
4. E. Balagurusamy – Programming in ANSI C
5. Narasimha Karumanchi – Data Structures and Algorithms Made Easy
6. Mark Allen Weiss – Data Structures and Algorithms in C++
7. Thomas H. Cormen, Leiserson, Rivest, Stein – Introduction to Algorithms (CLRS)
8. Steven S. Skiena – The Algorithm Design Manual
9. Robert Sedgewick, Kevin Wayne – Algorithms

Online Learning Resources:

1. <https://www.geeksforgeeks.org/data-structures/>
2. <https://www.geeksforgeeks.org/fundamentals-of-algorithms/>
3. <https://www.programiz.com/dsa>
4. <https://leetcode.com/explore/>
5. <https://www.hackerrank.com/domains/data-structures>

**Chairperson
Board of Studies (CSE)**

R24HS07	Quantitative Aptitude & Effectual Communication skills	0	0	2	1
(Common to All)					

Course Objectives:

1. **Develop problem-solving skills:** Students will be able to apply advanced concepts of Mensuration, Statistics, and Pure Mathematics to solve complex quantitative problems.
2. **Enhance data analysis skills:** Students will be able to Utilize principles of Permutations, Combinations, and Probability to estimate outcomes and interpret data from various charts.
3. **Develop effective communication skills:** Students will be able to Demonstrate effective verbal, non-verbal, and written communication skills suitable for professional environments.
4. **Enhance personal development skills:** Students will be able to Perform self-assessment and apply emotional intelligence and time management techniques for personal development.
5. **Develop leadership skills:** Students will be able to Exhibit leadership, teamwork, and conflict resolution skills during group interactions and collaborative tasks.

Course Objectives:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24HS07.1	Solve advanced problems in Mensuration, Statistics, and Pure Math.	3	---	---	---	---	---	---	---	---	---	---	---	---	L3	
R24HS07.2	Analyze data and estimate probabilities for decision making.	3	---	---	---	---	---	---	---	---	---	2	---	---	L3	
R24HS07.3	Communicate effectively through verbal, non-verbal, and written modes.	---	---	---	---	---	---	1	---	2	---	---	---	---	L2	
R24HS07.4	Apply self-awareness and EQ to manage time, stress, and goals.	---	---	---	---	---	---	---	---	3	---	2	---	---	L4	
R24HS07.5	Demonstrate interpersonal skills, leadership, and team collaboration.	---	---	---	---	---	---	---	---	3	---	1	---	---	L3	

SYLLABUS**UNIT-I: Quantitative computation****15 Hours**

Advanced concepts on – Mensuration-2D: basic formulas of Triangles, Square, Rectangle, Parallelogram, Rhombus, Trapezium and Circles and explain advanced problems on Mensuration-2D, Mensuration-3D: basic formulas of Cube, Cuboid, Cylinder, Sphere, Cone, Prism and Pyramid and explain advanced problems on Mensuration-3D, Statistics: basic concepts of Mean, Mode and Median and explain advanced problems on Statistics and Pure Math's: Geometry, Surds, Quadratic

Equations, Logarithms.

CO's-CO1

Self-Learning Topic: Mixture and Allegations

UNIT-II: Quantitative Estimation

15 Hours

Permutations & Combinations: basic formulas of Permutations & Combinations and Advanced Problems on linear and circular arrangement, problem based on with and without replacement, repetition and non-repetition, Probability: basic concepts of Probability & Advanced Problems on Coin based and Dice problems. Data Interpretation: Explain Advanced problems on Bar Chart, Pie Chart, Line Chart and Tabular Chart.

CO's-CO2

Self-Learning Topic: Algebra

UNIT-III: Communication Skills

12 Hours

Verbal Communication: Effective speaking and articulation, Active listening and paraphrasing, Providing and receiving constructive feedback. Non-Verbal Communication: Body language (gestures, posture, eye contact), Facial expressions and their interpretation, Dress code and professional appearance. Written Communication: Email Etiquette (tone, structure, clarity), Writing professional reports, notices, and memos, Clarity, coherence, and grammar basics for professional settings. Presentation Skills: Structuring a presentation (introduction, body, conclusion), Designing effective slides, managing stage fright and engaging the audience.

CO's-CO3

Self-Learning Topic: Active Listening Exercise

UNIT-IV: Self-Awareness & Personal Development

12 Hours

Self-Awareness: Performing a SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats), Understanding personal values and identifying goals. Emotional Intelligence (EQ): Defining EQ and its importance in the workplace, Self-management: regulating emotions and adapting to change, Empathy and understanding others' perspectives. Attitude and Motivation: Cultivating a positive attitude and a growth mindset, Developing self-confidence and assertiveness, Techniques for self-motivation and discipline. Time and Stress Management: Goal setting and prioritization (e.g., Eisenhower Matrix), Effective planning and avoiding procrastination, Stress management techniques and maintaining emotional well-being.

CO's-CO4

Self-Learning Topic: Stress Management Techniques

UNIT-V: Interpersonal and Team Skills

11 Hours

Teamwork and Collaboration: Defining a team versus a group, Roles within a team and stages of team formation, Effective contribution and shared responsibility. Conflict Resolution: Identifying sources of conflict, Strategies for managing and resolving workplace disagreements, Negotiation skills and achieving win-win outcomes. Leadership and Influence: Different styles of leadership., Delegation and guiding team members, The difference between formal and informal leadership. Group Discussions (GD): Techniques for initiating, contributing, and concluding a GD, Demonstrating listening and analytical skills during a discussion.

CO's-CO5

Self-Learning Topic: Role Plays on Conflict Techniques

Board of Studies : Management Studies

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Quantitative Aptitude for competitive examinations by R.S.Agrawal - S.Chand publications.

2. Business Communication by C.S.G Krishnamacharyulu & Lalitha Ramakrishna – Himalaya publishing house
3. Goleman, Daniel, Emotional Intelligence, Bantam Books.

Reference Books:

1. Numerical Aptitude and Reasoning Ability tests in Competitive exams by Prof.A.Balasubramanian.
2. Numerical Ability and Mathematical Aptitude by Dr.A.B.Rao
3. Communication skills and soft skills by E.Suresh Kumar , p.sreehari & j.savithri
4. Alex K., Soft Skills: Know Yourself & Know the World, S.Chand Publishing.

Web References:

1. <https://faceprep.in/logical> reasoning
2. www.sawaal.com
3. <http://nptel.ac.in/courses>
4. www.prepinsta.com

Model Questions from Quantitative Aptitude & Effectual Communication skills

L1: Remember

1. A cone and a sphere have the same radius of 18cm. If the cone and the sphere have the same volume, then what is the height (in cm) of the cone?
2. A metallic cuboid was processed by way of which its length, breadth and height got increased by 10%, decreased by 8% and increased by 5% respectively. What is the percentage (correct up to two decimal places) change in its volume?
3. If the perimeter of equilateral triangle is 72 cm, then what is its area in cm?
4. The perimeter of a regular hexagon with area $21\sqrt{3}$ cm² is:
5. The day-wise figures for the number of bags sold by a seller during a 30 - day period are as follows: 24, 26, 21, 25, 20, 35, 37, 45, 44, 10, 13, 15, 28, 27, 14, 34, 12, 16, 19, 21, 20, 23, 11, 24, 23, 28, 26, 29, 25, 20.
6. Find the median number of bags sold in a day for the given data?
7. A variable takes six values: 621, 622, 623, 625, 626, 627. Find the ratio between its standard deviation and mean?
8. What is the Mean Deviation of the first ten natural numbers?
9. Find the Mode of the given data:
10. 25, 66, 42, 52, 64, 25, 48, 52, 59, 52, 68, 25, 47, 52, 48, 66, 85, 26, 48, 66
11. If the Mode of a data is 12 and arithmetic mean is 6. then the median is?
12. What is the difference between the mean and the median of the given data?
13. 5, 9, 8, 15, 12, 9, 2, 19, 21, 11
14. What is the mean deviation (correct to three decimal places) of the number of days in a calendar month for the year 2020?
15. Calculate Mean – Mode + 2Median of the following data
16. 89, 92, 98, 99, 88, 86, 76, 88, 94, 96

L2: Understand

1. How many different words can be formed with the letters of the word "HARIYANA"?
2. From a group of 12 players a team of 5 players is formed in which one is captain and one is vice-captain. In how many different ways team can be formed?
3. How many different 6-digit numbers can be formed using 1,2,3,5,6,7,8 and 9 such that only

even digit comes at even places?

4. The letters of BOTTLE are permuted in all possible ways. How many of these words have all the vowels occupying even places?
5. Ashmita has 3 novels and 4 dictionaries. Determine the number of ways in which all 7 books can be arranged on the shelf, if the novels are on the left and the dictionaries are on the right?
6. A bag contains 4 strawberries and 8 grapes. If 2 fruits are drawn, what is the probability that both the fruits taken out from the bag are strawberries?
7. Carl has a standard deck of 52 cards. He pulls out three cards from the deck back to back. What is the probability that first two are king and the last card is a queen.
8. The probability of snowfall on Wednesday and Friday is 0.4 and 0.50 respectively. What is the probability of snowfall on both days?

L3: Apply

1. What is meant by verbal and non-verbal communication? Give two examples of each.
2. List the barriers to effective communication in a professional environment.
3. Define Emotional Intelligence (EQ) and mention its key components.
4. What are the essential elements that should be included in a professional resume?
5. Name any four types of leadership styles.
6. What is business etiquette? List any four commonly accepted workplace etiquette practices.

L4: Analyzing

1. Draft a professional email to your supervisor requesting approval for a one-day leave of absence.
2. Write a cover letter for the position of Software Trainee at an IT company, highlighting your technical and soft skills.
3. The manager asked the team to _____ their responsibilities clearly to avoid overlap during the project. (fill in the blank)
4. Prepare a short group discussion outline on the topic: 'Is work-life balance achievable in today's fast-paced world?'
5. Describe a time when you had to adapt quickly to a new work environment or assignment. What steps did you take?
6. Outline a one-week time management plan for a student preparing for campus placements.

L5: Evaluating

1. How would you differentiate between assertive and aggressive communication? Support with examples.
2. Explain how active listening contributes to effective interpersonal communication.
3. The new employee felt _____ because his team members did not acknowledge his contributions. (sentence completion)
4. Describe a situation where a team conflict could be resolved through negotiation rather than confrontation.
5. How does emotional intelligence help in managing stress and maintaining work-life balance?
6. Explain the importance of grooming and professional image in the workplace.

Chairperson
Board of Studies (MBA)

R24CSPC21**Artificial Intelligence and Machine Learning****3 0 0 3**

(Computer Science and Engineering)

Course Objectives:

1. To train the students to understand different types of AI agents.
2. To understand various AI search algorithms.
3. Fundamentals of knowledge representation, building of simple knowledge-based systems and to apply knowledge representation.
4. To introduce the basic concepts and techniques of machine learning and the need for Machine learning techniques for real world problem.
5. To provide understanding of various Machine learning algorithms and the way to evaluate the performance of ML algorithms.

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSPC 21.1	Understand the informed and uninformed problem types and apply search strategies to solve them.	3	3	1	1	1	-	-	-	-	-	2	3	1	L1, L2
R24CSPC 21.2	Apply difficult real life problems in a state space representation so as to solve those using AI techniques like searching and game playing.	3	3	2	2	1	-	-	-	1	-	2	3	2	L2, L3
R24CSPC 21.3	Apply machine learning techniques in the design of computer systems	3	3	3	2	2	-	-	-	1	-	2	3	3	L3, L4
R24CSPC 21.4	To differentiate between various categories of ML algorithms	3	2	1	1	1	-	-	-	-	-	2	3	1	L4, L5
R24CSPC 21.5	Design and make modifications to existing machine learning algorithms.	3	3	3	2	2	1	-	-	1	-	3	3	3	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction: AI problems, Agents and Environments, Structure of Agents, Problem Solving Agents Basic Search Strategies: Problem Spaces, Uninformed Search (Breadth-First, Depth-First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A*), Constraint Satisfaction (Backtracking, Local Search)

CO's–CO1**Self-Learning Topics:** Real-world applications of search algorithms (Google Maps, games)**UNIT II:****15 Hours****Advanced Search:** Constructing Search Trees, Stochastic Search, AO* Search Implementation,

Minimax Search, Alpha-Beta Pruning, Basic Knowledge Representation and Reasoning: Propositional Logic, First-Order Logic, Forward Chaining and Backward Chaining, Introduction to Probabilistic Reasoning, Bayes Theorem **CO's–CO2**

Self-Learning Topics: Game playing AI (Chess, Tic-Tac-Toe using Minimax), Bayesian Networks

UNIT III: 14 Hours

Machine-Learning : Introduction, Machine Learning Systems, Forms of Learning: Supervised and Unsupervised Learning, reinforcement – theory of learning – feasibility of learning – Data Preparation– training versus testing and split. **CO's–CO3**

Self-Learning Topics: Data preprocessing techniques (normalization, encoding), Scikit-learn library

UNIT IV: 13 Hours

Supervised Learning: Regression: Linear Regression, Multi Linear Regression, Polynomial Regression, Logistic Regression, Non-linear Regression, Model evaluation methods.

Classification: Support Vector Machines (SVM) , Naive Bayes classification **COs–CO4**

Self-Learning Topics: Real-world ML case studies (spam detection, fraud detection), ROC curves

UNIT V: 13 Hours

Unsupervised Learning: Nearest neighbor models, K-means, Clustering around medoids silhouettes, Hierarchical clustering, k-d trees, clustering trees, learning ordered rule lists, Learning unordered rule. **Reinforcement learning:** State and Action Spaces. **CO's–CO5**

Self-Learning Topics: Deep Learning, Applications of Reinforcement Learning (robotics, gaming)

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Russell, S. and Norvig, P, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2010.
2. MACHINE LEARNING An Algorithmic Perspective 2nd Edition, Stephen Marsland, 2015, by Taylor & Francis Group, LLC Introduction to Machine Learning, The Wikipedia Guide

Reference Books:

1. Artificial Intelligence, Elaine Rich, Kevin Knight, Shivasankar B. Nair, The McGraw Hill publications, Third Edition, 2009.
2. Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 6th ed., 2009.
3. Introduction to Machine Learning, Second Edition, Ethem Alpaydin, the MIT Press, Cambridge, Massachusetts, London, England.
4. Machine Learning, Tom M. Mitchell, McGraw-Hill Science, ISBN: 0070428077
5. Understanding Machine Learning: From Theory to Algorithms, c 2014 by ShaiShalev Shwartz and Shai Ben-David, Published 2014 by Cambridge University Press.

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--

L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Artificial Intelligence.
2. What is an intelligent agent?
3. Define problem space.
4. List types of search strategies.
5. What is BFS?
6. What is DFS?
7. Define heuristic function.
8. What is propositional logic?
9. Define supervised learning.
10. What is regression?
11. Define clustering.
12. What is reinforcement learning?

L2: Understand

1. Explain the concept of agents and environments.
2. Describe uninformed search strategies.
3. Explain informed search with an example.
4. Summarize A* algorithm.
5. Explain constraint satisfaction problems.
6. Describe propositional logic with example.
7. Explain forward and backward chaining.
8. Describe the concept of training vs testing data.
9. Explain supervised vs unsupervised learning.
10. Describe linear regression.
11. Explain K-means clustering.
12. Describe reinforcement learning basics.

L3: Apply

1. Apply BFS to solve a given problem.
2. Apply DFS to a search tree.
3. Use A* algorithm for path finding.
4. Solve a constraint satisfaction problem using backtracking.
5. Apply Bayes theorem to a probability problem.
6. Construct a truth table for a logical expression.
7. Apply forward chaining to derive conclusions.
8. Perform linear regression on a small dataset.
9. Apply logistic regression for classification.
10. Use Naive Bayes classifier on sample data.

11. Perform K-means clustering for given points.
12. Apply nearest neighbor method for classification.

L4: Analysing

1. Compare BFS and DFS in terms of efficiency.
2. Analyze the limitations of uninformed search.
3. Differentiate heuristic and non-heuristic search.
4. Analyze A* algorithm performance.
5. Compare propositional and first-order logic.
6. Analyze forward vs backward chaining.
7. Differentiate supervised and unsupervised learning.
8. Analyze regression vs classification techniques.
9. Compare SVM and Naive Bayes.
10. Analyze clustering techniques (K-means vs hierarchical).
11. Compare different distance measures in clustering.
12. Analyze reinforcement learning vs supervised learning.

L5: Evaluating

1. Evaluate BFS vs DFS for real-world applications.
2. Assess the effectiveness of heuristic search methods.
3. Evaluate A* algorithm in pathfinding problems.
4. Justify the use of constraint satisfaction methods.
5. Evaluate logical inference techniques.
6. Assess the performance of regression models.
7. Evaluate classification algorithms (SVM vs Naive Bayes).
8. Assess clustering algorithms for large datasets.
9. Evaluate K-means limitations.
10. Judge the suitability of ML algorithms for a given dataset.
11. Evaluate reinforcement learning in decision-making.
12. Assess advantages of unsupervised learning.

L6: Create

1. Design an intelligent agent for a real-world application.
2. Develop a search strategy for route optimization.
3. Design a heuristic function for a problem.
4. Construct a knowledge-based system using logic.
5. Design a rule-based expert system.
6. Develop a machine learning model for prediction.
7. Design a regression model for real-world data.
8. Create a classification system using SVM.
9. Design a clustering solution for customer segmentation.
10. Develop a recommendation system using ML.
11. Propose a reinforcement learning model for a game.
12. Design an AI system integrating multiple techniques.

R24CSPC22

Cryptography & Network Security
(Computer Science and Engineering)

3 0 0 3

Course Objectives:

1. To understand the fundamentals of security including security attacks services mechanisms and network Security concepts
2. To understand and apply public key cryptography hash functions and message authentication techniques for ensuring data security and integrity
3. To learn cryptographic hash functions and message authentication techniques including SHA and HMAC for ensuring data integrity and security
4. To learn electronic mail security techniques including S MIME for secure communication
5. To analyze IP security mechanisms including encapsulating security payload and Internet Key Exchange for secure data transmission

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSPC 22.1	Apply cryptography and security principles to analyze attacks, design systems	3	2	-	-	2	-	-	-	-	-	1	2	1	L1, L2
R24CSPC 22.2	Analyze classical ciphers and apply DES, AES algorithms to ensure strong, secure data encryption	3	3	2	1	3	-	-	-	-	-	1	3	2	L2, L3
R24CSPC 22.3	Apply RSA, SHA, HMAC techniques ensuring data security, integrity, authentication using public key cryptography principles	3	3	2	2	2	-	-	-	-	-	1	3	2	L3, L4
R24CSPC 22.4	Apply digital signatures, key management, authentication mechanisms ensuring secure communication and verified data authenticity	3	2	3	2	3	1	1	-	-	-	2	3	2 3	L4, L5
R24CSPC 22.5	Analyze network, internet security mechanisms including email, IP protocols to protect data during transmission securely	3	3	3	2	3	2	1	1	1	1	2	3	3	L5, L6

SYLLABUS
UNIT I:
12 Hours
Introduction to Security: Security Attacks, Security Services, Security Mechanisms, Fundamental Security Design Principles, Attack Surfaces and Attack Trees, Model for Network Security

Mathematics of Cryptography: Algebraic Structures (Groups, Rings, Fields, Galois Fields), Divisibility and the Division Algorithm, Euclidean Algorithm, Prime Numbers (basis of cryptography), Fermat's and Euler's Theorems (used in encryption) **CO's-CO1**

Self-Learning Topics: Testing for Primality, Chinese Remainder Theorem (optimization in computations), Discrete Logarithms (hard problem used for security)

UNIT II:

12 Hours

Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography

Block Ciphers: Traditional Block Cipher Structure, The Data Encryption Standard, The Strength of DES, Block Cipher Design Principles, Advanced Encryption Standard, AES Structure, AES Transformation Functions, AES Key Expansion, Block Cipher Modes of Operation **CO's-CO2**

Self-Learning Topics: Multiple Encryption and Triple DES, comparison of AES, DES, and modern ciphers

UNIT III:

15 Hours

Public-Key Cryptography: Principles of Public-Key Cryptosystems, The RSA Algorithm, Diffie Hellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Cryptography

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Requirements and Security, Secure Hash Algorithm (SHA)

Message Authentication Codes: Requirements for Message Authentication Codes, HMAC

CO's CO3

Self-Learning Topics: Implementation of RSA algorithm and Diffie Hellman key exchange, Study of Secure Hash Algorithm and HMAC for message authentication

UNIT IV:

13 Hours

Digital Signatures: Digital Signatures, Elgamal Digital Signature Scheme, Schnorr Digital Signature, Scheme Elliptic Curve Digital Signature Algorithm

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public-Key Infrastructure

User Authentication: Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Remote User-Authentication Using Asymmetric Encryption: **CO's-CO4**

Self-Learning Topics: Kerberos, NIST Digital Signature Algorithm digital signatures in blockchain systems, , biometric authentication systems.

UNIT V:

15 Hours

Network and Internet Security

Electronic Mail Security: Internet Mail Architecture, Email Formats, Email Threats and Comprehensive Email Security, S/MIME.

FIRE WALLS: The Need for Firewalls, Firewall Characteristics, Types of Firewalls, Firewall Configurations. **CO's-CO5**

Self-Learning Topics: SSL/TLS vs IPsec, VPN security, Email encryption tools

Board of Studies : Computer Science and Engineering

Approved in BoS No :03

Approved in ACM No :03

Text Book:

1. Cryptography and Network Security, William Stallings, 8th Edition, Pearson Education ,

Reference Books:

1. Cryptography, and Network Security and Cyber Laws, Bernard L. Menezes, Ravinder Kumar, Cengage Learning.
2. Cryptography and Network Security, Behrouz A Forouzan, Debdeep Mukhopadhyaya, 3rd Edition, Mc-GrawHill.
3. Network Security Illustrated, Jason Albanese, Wes Sonnenreich, McGrawHill.

Web References:

1. <https://nptel.ac.in/courses/106105162>
2. <https://nptel.ac.in/courses/106/105/106105031>
3. <https://www.geeksforgeeks.org/computer-networks/cryptography-tutorial/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define the primary goals of cryptography.
2. List and briefly describe various types of cryptographic attacks.
3. What is symmetric encryption? Name two examples.
4. Define the terms: ciphertext, plaintext, and key.
5. What is the difference between DES and AES?
6. List the steps in RSA key generation.
7. What is a cryptographic hash function?
8. Define firewall. List any two types.
9. What is the role of S/MIME in email security?

L2: Understand

1. Explain the working of the Advanced Encryption Standard (AES) with a block diagram
2. Describe the services and mechanisms in cryptography with examples.
3. Explain how the RSA algorithm performs encryption and decryption.
4. Describe the mathematical foundation behind the Elliptic Curve Cryptosystem (ECC).
5. Differentiate between symmetric and asymmetric encryption.
6. Explain the structure and working of SHA-1 and SHA-3.
7. Compare and contrast packet filtering and proxy firewalls.
8. Describe the process of key distribution using asymmetric encryption.
9. Explain how user authentication works using symmetric encryption.

L3: Apply

1. Encrypt a given message using the Caesar cipher with a shift of 3.

2. Apply the RSA algorithm to encrypt and decrypt a small numeric message.
3. Use DES to perform one round of encryption on a 64-bit block (assume input).
4. Demonstrate how to compute a message digest using SHA-256.
5. Implement a rule-based firewall configuration for blocking a specific IP address range.
6. Simulate public key distribution using a digital certificate model.
7. Apply the ElGamal cryptosystem on a sample message with given key parameters.
8. Show how to create an authentication protocol using nonce and symmetric encryption.
9. Apply the IKE protocol steps to establish a secure session.

L4: Analyze

1. Analyze the strength of DES and compare it with AES in terms of security and performance.
2. Compare RSA and ECC on key size, speed, and security.
3. Identify possible attacks on hash functions and their implications.
4. Distinguish between stateless and stateful firewalls with advantages and limitations.
5. Analyze the pros and cons of symmetric vs. asymmetric key management techniques.
6. Compare various types of cryptographic hash functions and their output size.
7. Examine the vulnerabilities of IP security without proper authentication headers.
8. Differentiate between the SHA-2 and SHA-3 algorithms in design and functionality.
9. Analyze a real-world scenario involving a failure of user authentication protocol.

L5: Evaluate

1. Evaluate the effectiveness of block ciphers versus stream ciphers in different applications.
2. Justify the use of public key cryptography in e-commerce security.
3. Assess the trade-offs between firewall complexity and network performance.
4. Evaluate the need for secure key distribution in wireless networks.
5. Critique the role of authentication protocols in cloud-based systems.
6. Recommend the best cryptographic techniques for securing IoT communications.
7. Evaluate the security level provided by combining AES with HMAC.
8. Appraise the relevance of Rabin cryptosystem in modern security.
9. Compare the role of authentication headers and encapsulating security payloads in IPSec.

L6: Create

1. Design a secure communication model combining symmetric and asymmetric encryption.
2. Create a custom authentication protocol for mobile banking using ECC.
3. Develop a security policy using firewalls, VPN, and intrusion detection systems for an enterprise.
4. Design a secure key distribution mechanism for a smart city IoT system.
5. Build a secure email system using S/MIME with digital signatures and encryption.
6. Construct a hybrid cryptographic solution using RSA and AES for cloud storage.
7. Create a secure file-sharing mechanism using hash verification and public key infrastructure.
8. Design a role-based firewall system for a university campus.

**Chairperson
Board of Studies (CSE)**

R24HS08 MANAGERIAL ECONOMICS AND FINANCIAL ACCOUNTING 2002

(Common to EEE, Mechanical, ECE, CSE, CSE- AI&ML, CSE-D)

Course Objectives:

1. To inculcate the basic knowledge of microeconomics and financial accounting
2. To make the students learn how demand is estimated for different products, input-output relationship for optimizing production and cost
3. To Know the Various types of market structure pricing methods and strategy
4. To give an overview of investment appraisal methods to encourage the students to learn
5. how to plan long-term investment decisions.
6. To provide fundamental skills in accounting and to explain the process of preparing
7. financial statements.

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs												Dok	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	POS1		PSO2
R24HS05	Understand what Managerial Economics is and how demand and its factors affect business decisions.	3	2	-	-	2	-	-	-	2	2		-	-	L1, L2
R24HS05	Learn how to predict future demand and understand how production and costs work in a business.	3	3	2	3	-	-	-	-	2	2	2	-	-	L2, L3
R24HS05	Know different types of markets and how companies decide prices in various situations.	3	3	2	2	2	-	-	-	2	2	2	-	-	L3, L4
R24HS05	Understand types of business organizations, business cycles, and how to plan for long-term investments.	2	2	-	-	3	-	-	-	3	3		-	-	L4, L5
R24HS05	Learn how to prepare and read financial statements and check a company's financial health using simple tools.	2	2	-	2	3	-	-	2	3	3	3	-	-	L5, L6

SYLLABUS**UNIT I****10 Hours**

Managerial Economics and Demand Analysis: Definition, nature, and scope of Managerial Economics – Relationship with other disciplines (Accounting, Finance, Behavioral Economics) – Basic economic tools: Opportunity Cost, Marginal Analysis, Discounting Principle – Demand:

Meaning, types, determinants – Law of Demand and its exceptions – Elasticity of Demand: Types and measurement (Total Outlay, Point, Arc Methods) – Significance in pricing decisions.

CO's-CO1

Self-Learning Topics

Study how managerial economics supports business decision-making in modern organizations. Analyze real-world examples of demand fluctuations and examine how elasticity of demand influences pricing strategies in different industries.

UNIT II

8 Hours

Demand Forecasting and Production-Cost Analysis: Demand Forecasting: Meaning, factors, methods (Survey, Statistical, Delphi) – Forecasting for new products – Characteristics of a good forecasting method – Production Function: Isoquants, Isocosts, MRTS – Cobb-Douglas, CES functions – Cost concepts: Fixed vs. Variable, Explicit vs. Implicit, Opportunity Cost – Break-even analysis: Meaning, graphical representation, managerial significance

CO's-CO2

Self-Learning Topics

Explore various demand forecasting techniques used by organizations and compare their effectiveness in predicting market demand. Prepare a simple break-even analysis for a business scenario and interpret its managerial significance.

UNIT III

12Hours

Market Structures and Pricing Policies Market Structures: Features of Perfect Competition, Monopoly, Monopolistic Competition, Oligopoly – Price-output determination under different market conditions – Pricing Methods: Cost-based, Marginal Cost, Value-based, Skimming, Penetration, Bundling, Surge Pricing – Internet Pricing Models: Flat Rate, Usage-based, Smart Market Mechanism.

CO's-CO3

Self-Learning Topics

Investigate different market structures and identify businesses operating under each type of market condition. Study modern pricing strategies such as dynamic pricing, surge pricing, and online pricing models used by digital platforms and e-commerce companies.

UNIT IV

12 Hours

Industrial Organization and Capital Budgeting: Forms of Business Organization: Sole Proprietorship, Partnership, Joint Stock Company, Co-operative Societies, LLPs – Public Enterprises: Types and features – Business Cycles: Phases and characteristics – Capital Budgeting: Need, importance – Methods: Payback Period, ARR, IRR, NPV, Profitability Index (simple problems).

CO's-CO4

Self-Learning Topics

Analyze the advantages and limitations of different forms of business organizations in India. Prepare simple capital budgeting calculations using Payback Period, ARR, NPV, and IRR methods and interpret their usefulness in investment decision-making.

UNIT V

12Hours

Financial Accounting and Analysis: Accounting Principles – Double-entry system – Journal, Ledger, Trial Balance, Final Accounts (with simple adjustments) – Introduction to Tally/ERP tools – Limitations of Financial Statements – Financial Statement Analysis: Liquidity, Profitability, Solvency Ratios – Fund Flow Statement – Cash Flow Statement (basic concepts as per AS-3).

CO's-CO5

Self-Learning Topics

Explore the preparation and interpretation of financial statements using basic accounting principles. Study the importance of ratio analysis, fund flow, and cash flow statements in evaluating the financial performance and liquidity position of organizations.

Textbooks:

1. Varshney & Maheswari: Managerial Economics, Sultan Chand.
2. Aryasri: Business Economics and Financial Analysis, 4/e, MGH.

Reference Books:

1. Ahuja Hl Managerial economics Schand.
2. S.A. Siddiqui and A.S. Siddiqui: Managerial Economics and Financial Analysis, New Age International.
3. Joseph G. Nellis and David Parker: Principles of Business Economics, Pearson, 2/e, New Delhi.
4. Domnick Salvatore: Managerial Economics in a Global Economy, Cengage.

Web References:

1. <https://www.slideshare.net/123ps/managerial-economics-ppt>
2. <https://www.slideshare.net/rossanz/production-and-cost-45827016>
3. <https://www.slideshare.net/darkyla/business-organizations-19917607>
4. <https://www.slideshare.net/balarajbl/market-and-classification-of-mark>
5. <https://www.slideshare.net/ruchi101/capital-budgeting-ppt-59565396>
6. <https://www.slideshare.net/ashu1983/financial-accounting>

Internal Assessment Pattern

Cognitive Level	InternalAssessment#1(%)	InternalAssessment#2(%)
L1	17	-
L2	17	-
L3	33	30
L4	33	30
L5	-	28
L6	-	14
Total (%)	100	100

Sample Short and Long Answer s to Question so Various Cognitive Levels**Short Answer Questions:**

1. Define Managerial Economics. (L1)
2. What is the Law of Demand? (L1)
3. List any three determinants of demand. (L1)
4. Explain the relationship between Managerial Economics and other disciplines. (L2)
5. Differentiate between the Total Outlay Method and the Arc Method. (L2)

Long Answer Questions:

1. Explain the types and significance of elasticity of demand. (L2)
2. Analyze the role of elasticity in business decision-making. (L4)

Unit II: Demand Forecasting & Production-Cost Analysis**Short Answer Questions:**

1. Define Demand Forecasting. (L1)
2. List two qualitative forecasting methods. (L1)
3. What is the difference between Isoquant and Isocost? (L2)

4. Classify costs into fixed and variable types. (L3)
5. Identify the break-even point from the given data. (L3)

Long Answer Questions:

1. Describe demand forecasting methods and explain one in detail. (L2)
2. Draw isoquants and isocosts; explain MRTS. (L3)
3. Analyze managerial implications of break-even analysis. (L4)

Unit III: Market Structures & Pricing Policies

Short Answer Questions:

1. Define Perfect Competition. (L1)
2. List features of Monopolistic Competition. (L1)
3. Explain Penetration Pricing. (L2)
4. Compare Perfect Competition and Monopoly. (L4)
5. Suggest a pricing strategy for a new e-commerce service. (L6)

Long Answer Questions:

1. Explain various pricing methods with examples. (L2)
2. Evaluate Smart Market Mechanism in digital services. (L5)
3. Analyze the Indian retail sector's market structure. (L4)

Unit IV: Industrial Organization & Capital Budgeting

Short Answer Questions:

1. List forms of business organization. (L1)
2. What are the features of a Joint Stock Company? (L2)
3. Define Capital Budgeting. (L1)
4. Identify suitable capital budgeting methods for short-term projects. (L3)
5. Differentiate between ARR and IRR. (L2)

Long Answer Questions:

1. Compare business forms and their startup suitability. (L4)
2. Explain Payback, NPV, and IRR with examples. (L2)
3. Evaluate a capital investment using the Profitability Index. (L5)

Unit V: Financial Accounting & Analysis

Short Answer Questions:

1. What is Double-entry Accounting? (L1)
2. Define Trial Balance and its purpose. (L1)
3. Explain the limitations of financial statements. (L2)
4. Prepare a Trial Balance from given transactions. (L3)
5. Identify liquidity ratios and explain their use. (L2)

Long Answer Questions:

1. Prepare Final Accounts from a Trial Balance. (L3)
2. Analyze financial health using ratios. (L4)
3. Create a Fund Flow Statement using data. (L6)

**Chairperson
Board of Studies (MBA)**

Course Objectives:

1. To understand the fundamentals of cloud computing and virtualization.
2. To learn cloud architecture and service models like IaaS, PaaS, and SaaS.
3. To understand cloud services, scaling, and disaster recovery mechanisms.
4. To gain knowledge of cloud application platforms and development tools.
5. To explore real-world cloud computing systems and industry applications.

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSOE 10.1	Understand the concept of virtualization and how this has enabled the development of Cloud Computing.	3	2	1	-	2	-	-	-	-	-	-	3	2	L1, L2
R24CSOE 10.2	Explain cloud architecture and different service models	3	3	2	-	1	-	-	-	-	-	-	2	3	L2, L3
R24CSOE 10.3	Analyze scaling, security, and disaster recovery in cloud systems.	3	3	2	2	2	-	-	-	-	1	-	2	3	L3, L4
R24CSOE 10.4	Design simple cloud-based applications using platform tools.	2	2	3	2	3	-	-	-	-	1	2	3	3	L4, L5
R24CSOE 10.5	Identify and evaluate real-world cloud platforms and applications.	2	2	1	1	3	-	-	-	-	1	2	2	2	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction to Cloud Computing & Virtualization Basics of Cloud Computing, what is Cloud Computing, Vision and need of cloud, Basic cloud characteristics, Benefits of cloud computing Virtualization Basics Introduction to virtualization, Virtual machines (VM concept), Basic types of virtualizations, Introduction to hypervisors, Simple tools: VMware Microsoft Hyper-V

CO's –CO1

Self-Learning Topics: Evolution of Cloud Computing (history and development)

UNIT II:**13 Hours**

Cloud Architecture: Cloud Basics, Cloud architecture Overview, Cloud service models: IaaS, PaaS, SaaS Types of Cloud: Public cloud, Private cloud, Hybrid cloud, Basic Concepts: Scalability, Fault tolerance, Introduction to cloud security

CO's –CO2

Self-Learning Topics: Micro services vs. Monolithic architecture (basic idea)

UNIT III:**14 Hours**

Cloud Services & Disaster Recovery: Cloud Services Basics, Storage as a Service, Database as a Service Software/Process as a Service, Scaling Basics: What is scaling, Basic capacity planning, Disaster Recovery: Backup concept, Disaster recovery meaning, Importance of data recovery

CO's–CO3

Self-Learning Topics: Real-time backup systems in cloud (Google backup, AWS backup basics)

UNIT IV:

15 Hours

Cloud Application Platform (Aneka Basics): Platform Overview, Introduction to Aneka, Basic idea of cloud application Platforms, Basic Components: Container concept, Simple service layers, Introduction to SDK Deployment Models: Private cloud, public cloud, Hybrid cloud

CO's –CO4

Self-Learning Topics: Cloud SDK usage and simple application deployment concepts

UNIT V:

13 Hours

Cloud Applications & Industry Basics: Cloud Applications, Basic idea of cloud applications, Scientific applications (health, biology basics), Business applications (CRM, ERP basics), Cloud Platforms, Introduction to: Amazon Web Services, Google App Engine, and Microsoft Azure: Basic Idea, what cloud platforms do, Simple usage of cloud services.

CO's- CO5

Self-Learning Topics: Server fewer computing basics (Function as a Service concept)

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Book:

1. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, and S.Thamarai Selvi from TMH 2013.
2. George Reese Cloud Application Architectures, First Edition, O'Reilly Media 2009.

Reference Books:

1. Cloud Computing and SOA Convergence in Your Enterprise A Step-by-Step Guide by David S. Linthicum from Pearson 2010.
2. Cloud Computing 2nd Edition by Dr. Kumar Saurabh from Wiley India 2012.
3. Cloud Computing – web-based Applications that change the way you work and collaborate Online – Michael Miller. Pearson Education.

Internal Assessment Pattern

Sample Short and Long Answers questions of Various Cognitive Levels

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

L1: Remember

1. Define Cloud Computing.
2. What is virtualization?
3. What is a Virtual Machine (VM)?
4. List the basic characteristics of cloud computing.
5. What are the benefits of cloud computing?
6. List the cloud service models.
7. Define IaaS, PaaS, and SaaS with one example each.
8. List the types of cloud deployment models.
9. What is scaling in cloud computing?
10. Define disaster recovery in cloud computing.
11. What is serverless computing / Function as a Service (FaaS)?

L2: Understand

1. Explain the concept of virtualization and its role in cloud computing.
2. Describe the key characteristics of cloud computing.
3. Explain cloud computing architecture with a neat diagram idea.
4. Compare IaaS, PaaS, and SaaS with examples.
5. Explain public, private, and hybrid cloud with differences.
6. Explain scalability in cloud computing with types.
7. Describe fault tolerance and its importance in cloud systems.
8. Explain disaster recovery and backup strategies in cloud computing.
9. Explain cloud application platforms and their components (like container and services).
10. Explain cloud applications in business and scientific domains.
11. Describe serverless computing (FaaS) and its advantages in cloud systems.

L3: Apply

1. How would you use virtualization to run multiple operating systems on a single physical machine? Explain with a real scenario.
2. A company wants to reduce hardware cost. How can cloud computing help them achieve this?
3. A startup needs scalable web hosting. Which cloud service model would you choose (IaaS, PaaS, SaaS)? Justify your answer.
4. A hospital wants secure and private data storage. Which cloud deployment model is suitable? Explain.
5. Design a simple cloud architecture for an online shopping website.
6. A video streaming platform is experiencing high traffic. Which scales method would you apply and why?
7. How would you design a disaster recovery plan for a banking system using cloud computing?
8. A company stores sensitive data in the cloud. What security measures would you apply to protect it?
9. How would you deploy a simple distributed application using a cloud platform like Aneka? Explain steps.
10. A university wants to move its ERP system to the cloud. How can cloud computing support this transformation?
11. How would you use server less computing (FaaS) to build a simple event-driven application?

L4: Analysing

1. Analyze how virtualization improves resource utilization in cloud computing environments.

2. Compare physical computing and virtualized cloud environments in terms of performance, cost, and scalability.
3. Analyze the differences between IaaS, PaaS, and SaaS in terms of control, flexibility, and usage.
4. Compare public, private, and hybrid cloud models and analyze their suitability for different organizations.
5. Analyze the role of cloud architecture in ensuring scalability and reliability.
6. Compare vertical and horizontal scaling and analyze which is better for large-scale cloud applications.
7. Analyze the importance of fault tolerance in cloud systems with real-world examples.
8. Evaluate different disaster recovery strategies and analyze their effectiveness in critical systems like banking or healthcare.
9. Analyze the role of cloud application platforms (like Aneka) in simplifying distributed application development.
10. Analyze the advantages and limitations of serverless computing compared to traditional cloud deployment models.
11. Compare AWS, Google Cloud, and Microsoft Azure and analyze their strengths in different application areas.

L5: Evaluating

1. Evaluate the advantages and limitations of virtualization in cloud computing.
2. Justify whether cloud computing is better than traditional computing for enterprises.
3. Evaluate IaaS, PaaS, and SaaS and recommend which is best for a startup company.
4. Justify the selection of public, private, or hybrid cloud for a government organization.
5. Evaluate the importance of cloud architecture in ensuring system reliability and performance.
6. Evaluate vertical vs horizontal scaling and justify which is more efficient for modern cloud systems.
7. Assess the effectiveness of different cloud security techniques in protecting sensitive data.
8. Evaluate disaster recovery strategies and justify the best approach for a financial system.
9. Evaluate the role of cloud application platforms (like Aneka) in simplifying distributed computing compared to traditional methods.
10. Evaluate AWS, Google Cloud, and Microsoft Azure based on performance, cost, and features.
11. Justify the use of serverless computing over traditional cloud deployment for event-driven applications.

L6: Create

1. Design a virtualized system for a college lab where multiple OS environments are required on the same hardware.
2. Propose a cloud-based solution to reduce infrastructure cost in a small organization using virtualization.
3. Design a cloud architecture for an e-commerce application explaining service models (IaaS, PaaS, SaaS).
4. Propose a hybrid cloud model for a healthcare system and justify your design.
5. Design a scalable cloud system architecture ensuring fault tolerance and security.
6. Design a disaster recovery plan for a banking system using cloud technologies.
7. Propose a scalable architecture for a video streaming platform handling millions of users.
8. Design a secure cloud storage system for sensitive government data.

9. Design and outline steps to develop a distributed cloud application using a platform like Aneka.
10. Design a cloud-based ERP system for a university using modern cloud platforms.
11. Propose a serverless computing-based application for a real-world event-driven system (e.g., online ticket booking or notification system)

**Chairperson
Board of Studies (CSE)**

R24CSPE07 AUGMENTED REALITY AND VIRTUAL REALITY 3 0 0 3
(Open Elective Course-II)

Course Objectives:

1. To understand the fundamentals and concepts of Augmented Reality (AR) and Virtual Reality (VR).
2. To analyze AR/VR hardware, devices, and interaction techniques.
3. To understand 3D visualization, rendering, and tracking technologies.
4. To illustrate AR/VR development tools and application development.
5. To understand the applications, challenges, and future trends in AR and VR technologies.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs												Dok	
		P01	P02	P03	P04	P05	P06	P07	P08	P09	P010	P011	PSO1		PSO2
R24CSPE 07.1	Describe operating system overview, functions, and services	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2
R24CSPE 07.2	Analyze process concept, inter-process communication, and synchronization	3	3	-	2	3	-	-	2	-	-	2	3	2	L2, L3
R24CSPE 07.3	Analyze process synchronization mechanisms and dead lock handling techniques.	3	3	2	2	3	-	-	-	2	-	2	3	3	L3, L4
R24CSPE 07.4	Describe methods for mapping secondary storage management.	3	2	2	2	3	-	-	-	-	-	2	3	2	L4, L5
R24CSPE 07.5	Evaluate protection mechanism and security	3	2	3	2	3	2	2	-	-	2	3	-	-	L5, L6

SYLLABUS**UNIT I****15 Hours**

Introduction to Augmented Reality and Virtual Reality: Definitions, History and Evolution of AR/VR, Characteristics of AR and VR, Components of AR/VR Systems, Types of VR Systems, Applications of AR/VR. Human Perception and Interaction: Human Vision, Depth Perception, Motion Tracking, User Interaction Techniques, Challenges in AR/VR Systems. **CO's–CO1**

Self-Learning Topics: Recent Trends and Innovations in AR and VR Technologies.

UNIT II**15 Hours**

AR/VR Hardware and Devices: Head Mounted Displays (HMDs), Sensors, Cameras, Tracking Devices, Input and Output Devices, Wearable Technologies. Interaction Techniques: Gesture Recognition, Voice Interaction, Haptic Devices, Motion Controllers, Eye Tracking Techniques. **CO's–CO2**

Self-Learning Topics: AR/VR Hardware Platforms and Commercial Devices.

UNIT III

15 Hours

3D Modeling and Rendering: 3D Graphics Fundamentals: Coordinate Systems, Transformations, 3D Modeling Concepts, Computer Graphics Basics. Rendering and Tracking: Rendering Techniques, Lighting and Shading, Marker-Based Tracking, Markerless Tracking, Simultaneous Localization and Mapping (SLAM), Object Recognition Techniques. **CO's–CO3**

Self-Learning Topics: Advanced Rendering Techniques and Real-Time Tracking.

UNIT IV

15 Hours

AV/VR Development and Applications: Development Platforms and Tools: Unity 3D, Unreal Engine, AR Core, AR Kit, Vuforia SDK. AR/VR Application Development: Mobile AR Applications, VR Simulations, Educational Applications, Gaming Applications, Healthcare and Industrial Applications. **COs–CO4**

Self-Learning Topics: Cross-Platform AR/VR Application Development.

UNIT V

10 Hours

Future Trends and Challenges in AR/VR: Challenges in AR/VR: Technical Challenges, User Experience Issues, Privacy and Security Concerns, Ethical Issues. Future Trends: Metaverse Concepts, Mixed Reality, Extended Reality (XR), AI Integration with AR/VR, Future Scope of AR and VR Technologies. **CO's–CO5**

Self-Learning Topics: Metaverse and Immersive Technologies

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. AR and VR Applications in Gaming and Entertainment Industries. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Industrial Applications of AR/VR in Manufacturing, Healthcare, and Education. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Metaverse, Mixed Reality, and Future Scope of Immersive Technologies. CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Augmented Reality: Principles and Practice – Dieter Schmalstieg and Tobias Hollerer.
2. Virtual Reality – Steven M. LaValle.
3. Learning Virtual Reality – Tony Parisi

Reference Books:

1. Understanding Virtual Reality – William Sherman and Alan Craig.
2. Practical Augmented Reality – Steve Aukstakalnis.
3. Developing Virtual Reality Applications – Alan B. Craig.
4. Augmented Reality for Developers – Jonathan Linowes.

Web References:

1. <https://nptel.ac.in/>
2. <https://learn.unity.com/>
3. <https://developer.apple.com/augmented-reality/>
4. <https://developers.google.com/ar>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define Augmented Reality.
2. What is Virtual Reality?
3. Define Head Mounted Display (HMD).
4. What is Motion Tracking?
5. Define Marker-Based Tracking.
6. What is SLAM?
7. Define Haptic Technology.
8. What is Mixed Reality?
9. Define Metaverse.
10. What is XR?

L2: Understand

1. Explain the difference between AR and VR.
2. Describe the components of AR/VR systems.
3. Explain the role of sensors in AR/VR devices.
4. Describe gesture recognition techniques.
5. Explain rendering techniques in AR/VR.
6. Describe the applications of AR/VR in education.
7. Explain the importance of tracking systems.
8. Describe privacy concerns in AR/VR systems.

L3: Apply

1. Demonstrate the use of AR Core in mobile applications.
2. Apply tracking techniques in an AR environment.
3. Develop a simple VR simulation using Unity 3D.
4. Apply rendering techniques to improve visual quality.
5. Implement gesture-based interaction in a VR application.
6. Develop a simple AR application for educational purposes.

L4: Analysing

1. Analyze the differences between marker-based and markerless tracking.
2. Compare ARCore and ARKit development platforms.
3. Analyze the impact of latency in VR systems.
4. Compare different interaction techniques in AR/VR.
5. Analyze challenges in implementing immersive technologies.
6. Examine the role of AR/VR in industrial applications.

L5: Evaluating

1. Evaluate the effectiveness of AR in education systems.
2. Assess the impact of VR in healthcare applications.
3. Evaluate different tracking techniques used in AR systems.
4. Assess the role of immersive technologies in smart industries.
5. Evaluate privacy and ethical issues in AR/VR.
6. Assess the future scope of Metaverse technologies.

L6: Create

1. Design a mobile AR application for campus navigation.
2. Develop a VR-based training simulator for industries.
3. Create an AR educational application using Unity 3D.
4. Design a gesture-controlled immersive VR environment.
5. Develop an AR/VR solution for healthcare rehabilitation.

**Chairperson
Board of Studies (CSE)**

R24CSPE08**CYBER SECURITY****3 0 0 3**

(Professional Elective-II)

Course Objectives:

1. To introduce the fundamental concepts of cyber security, including cyber threats, cybercrimes, and the need for security policies.
2. To provide knowledge of web security principles and network vulnerabilities, along with techniques to identify and prevent attacks such as SQL injection, XSS, and DDoS.
3. To develop understanding of security threats, malware, and essential protection mechanisms like authentication, access control, firewalls, and intrusion detection systems.
4. To explain the working and importance of intrusion detection systems (IDS), including different types, tools, and honeypots for threat detection.
5. To familiarize students with cyber laws, digital forensics, and legal aspects related to cyber-crimes, including the Indian IT Act.

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSPE 08.1	Understand basic cyber security concepts, threats, and tools.	3	2	–	–	3	2	–	2	–	–	–	3	2	L1, L2
R24CSPE 08.2	Analyze web security issues and network attack techniques.	3	3	2	2	3	–	–	–	–	–	–	3	3	L2, L3
R24CSPE 08.3	Identify security threats and apply suitable counter measures.	3	3	2	2	3	2	–	2	–	–	–	3	3	L3, L4
R24CSPE 08.4	Explain intrusion detection systems and related tools.	2	3	2	3	3	–	–	–	–	–	–	2	2	L4, L5
R24CSPE 08.5	Understand cyber laws and digital forensics fundamentals.	2	2	–	2	2	3	–	3	–	2	–	2	2	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction to Cyber Security: What is Cyber Security, its need, cyber-threats, Cyber Warfare-Cyber Crime-Cyber Terrorism-Cyber Espionage), Career Growth, Statistics, Inferences, Need for a Comprehensive Cyber Security Policy, Classification of Cyber Crimes, kinds of cyber-crimes, Reasons for Cyber Crimes, Cyber Security Tools: Nmap, Metasploit, Wireshark, Tcpdump, Snort.

CO's–CO1**Self-Learning Topics:** Cyber security awareness, social engineering, cyber stalking**UNIT II:****15 Hours**

Web security: Same origin Policy, Cross Origin Resource Sharing, DDOS, SQL Injection, XSS, Homograph, Generating and storing session tokens.

Networking Scanning & Security Measures: Packet Sniffing and spoofing, Network scanning types, port scanning & its tools, and Network Architecture **CO's–CO2**

Self-Learning Topics: Cross-Site Request Forgery (XSRF/CSRF), spear phishing.

UNIT III:

15 Hours

Security Threats and Vulnerabilities: Malware, Types of malwares, Overview of Security Threats and Vulnerability: Types of attacks on Confidentiality, Integrity and Availability, Security Counter-Measures: Passwords, Access Control (Authenticating users, Handling User Access), Antivirus Software, Firewalls, Intrusion Detection Systems & IPS. **CO's–CO3**

Self-Learning Topics: Types of firewalls, advanced persistent threat, Buffer Overflow

UNIT IV:

15 Hours

Intrusion Detection System: Intrusion Detection System Intruders, Intrusion Detection, Analysis Approaches, Network-Based IDS, Host-Based IDS, signature-based IDS, anomaly-based IDS, advantages and disadvantages of NIDS and HIDS Intrusion Detection Tools, snort architecture, snort rules, case studies of intrusion detection systems, Intrusion detection exchange format. Honeypots, different types of honeypots, benefits and dangers of honeypots. **COs–CO4**

Self-Learning Topics: firewall vs IDS, Physical IDS, honeynet

UNIT V:

13 Hours

Cyber Laws and Digital Forensics:

Digital Forensics: Introduction to Digital Forensics, historical background of digital forensics, Forensic Software, and Hardware, need for computer forensics science, special tools and techniques digital forensic life cycle, challenges in digital forensic.

Law Perspective: Introduction to the Legal Perspectives of Cybercrimes and Cybersecurity, Cybercrime and the Legal Landscape around the World, Why Do We Need Cyber laws, The Indian IT Act, Cybercrime Scenario in India, Digital Signatures and the Indian IT Act **CO's–CO5**

Self-Learning Topics: Cybercrime and Punishment

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No :03

Text Books:

1. William Stallings, Lawrie Brown, Computer Security Principles and Practice Third Edition, 2015
2. Nihad A. Hassan, Rami Hijazi, Digital Privacy and Security Using Windows: A Practical Guide, Apress, 2017
3. Sunit Belapure and Nina Godbole, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley India Pvt. Ltd, 2011.

Reference Books:

1. Michael T. Goodrich and Roberto Tamassia, Introduction to Computer Security, Addison Wesley, 2011.
2. Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed, 3rd Edition, John Wiley & Sons Inc, 2023.
3. Kozierok, Charles M. The TCP/IP guide: a comprehensive, illustrated Internet protocols reference, 2005
4. Harris, Shon, "CISSP all-in-one exam guide." Sixth edition (2013)

Web References:

1. <https://www.nist.gov/cyberframework>
2. <https://www.cisa.gov>
3. <https://owasp.org>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define Cyber Security.
2. List types of cyber threats.
3. What are tools like Nmap and Wireshark?
4. Define Same Origin Policy and CORS.
5. What is SQL Injection and XSS?
6. Define malware and list its types.
7. What are CIA triad components?
8. Define IDS and its types.
9. What is a honeypot?
10. Define digital forensics.
11. What is the Information Technology Act 2000?

L2: Understand

1. Explain the need for cyber security in modern systems.
2. Describe differences between cyber crime, cyber warfare, and cyber terrorism.
3. Explain how Same Origin Policy works.
4. Describe DDoS attacks and their impact. Explain the difference between a process and a thread.
5. Explain types of attacks on confidentiality, integrity, and availability.
6. Describe working of firewalls and antivirus.
7. Explain Snort architecture.
8. Describe signature-based vs anomaly-based IDS.
9. Explain digital forensic life cycle.
10. Describe need for cyber laws.

L3: Apply

1. Demonstrate how Metasploit can be used in vulnerability testing.

2. Classify a given scenario into types of cyber crimes.
3. Demonstrate how to detect SQL Injection vulnerability.
4. Apply packet sniffing using Tcp dump.
5. Apply access control techniques in a system.
6. Demonstrate password security mechanisms.
7. Write basic Snort rules.
8. Demonstrate how a honeypot detects attackers.
9. Apply forensic tools in investigating cyber crime.
10. Demonstrate evidence collection process.

L4: Analysing

1. Analyze reasons behind increasing cyber crimes.
2. Compare different cyber security tools and their functions.
3. Analyze differences between XSS and CSRF attacks.
4. Compare network scanning techniques.
5. Analyze different malware behaviors.
6. Compare IDS and IPS systems.
7. Analyze advantages and disadvantages of NIDS vs HIDS.
8. Compare different IDS approaches.
9. Analyze challenges in digital forensics.
10. Compare cyber laws across countries.

L5: Evaluating

1. Evaluate the need for a comprehensive cyber security policy in organizations.
2. Assess the impact of cyber espionage on national security.
3. Evaluate effectiveness of session token security mechanisms.
4. Assess risks of packet sniffing and spoofing.
5. Evaluate effectiveness of firewall protection.
6. Assess different authentication techniques.
7. Evaluate effectiveness of intrusion detection systems.
8. Assess risks involved in using honeypots.
9. Evaluate effectiveness of Indian cyber laws.
10. Assess legal implications of cyber crimes.

L6: Create

1. Design a basic cyber security awareness plan for an organization.
2. Propose a strategy to prevent social engineering attacks.
3. Design a secure web application architecture.
4. Create a plan to prevent DDoS attacks.
5. Design a secure authentication system.
6. Propose a defense mechanism against advanced persistent threats.
7. Design an IDS-based network security model.
8. Create a honeynet setup for research

**Chairperson
Board of Studies (CSE)**

R24CSPE09**DEVOPS****3 0 0 3**

(Professional Elective-II)

Course Objectives:

The objectives of this course for the student are to:

1. Learn why automation, culture, and metrics are essential to a successful DevOps project.
2. Learn how DevOps can positively impact your business's bottom line. Learn which major Companies are successfully utilizing
3. DevOps in their own engineering processes.
4. DevOps improves collaboration and productivity by automating infrastructure and
5. Workflow and continuously measuring applications performance.

Course Outcomes

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSP E09.1	CO1: Enumerate the principles of continuous development and deployment, automation of configuration management, inter-team collaboration, and IT service agility	2	2	1	1	1	1	-	1	2	1	-	2	1	L1, L2
R24CSP E09.2	CO2: Describe DevOps & DevSecOps methodologies and their key concepts	2	2	1	1	1	2	1	2	2	1	-	2	1	L2, L3
R24CSP E09.3	CO3: Illustrate the types of version control systems, continuous integration tools.	2	2	2	1	3	1	-	1	2	1	-	3	2	L3, L4
R24CSP E09.4	CO4: Continuous monitoring tools, and cloud models and set up complete private infrastructure using version control systems.	2	2	2	2	3	1	1	1	2	1	-	3	3	L4, L5
R24CSP E09.5	CO5: CI/CD tools to Know about DevOps maturity model.	2	3	3	2	2	2	1	2	3	2	1	3	3	L5, L6

SYLLABUS**UNIT I:****14 Hours**

Phases of Software Development life cycle. Values and principles of agile software development. Software Development Life Cycle (SDLC) models (Waterfall, Agile, Spiral, DevOps lifecycle), Agile principles (Scrum, Kanban), Limitations of traditional models

CO's-CO1**Self-Learning Topics:** Need for DevOps culture, Foundation for continuous development

UNIT II: **15 Hours**
 Fundamentals of DevOps: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system, Infrastructure as Code (IaC), DevOps ecosystem tools (Git, Jenkins, Docker, Kubernetes) **CO's–CO2**

Self-Learning Topics: Understanding DevOps workflow

UNIT III: **15 Hours**
 DevOps adoption in projects: Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes, Automation strategies, CI tools (Jenkins, GitHub Actions) **CO's–CO3**

Self-Learning Topics: Real-world DevOps implementation

UNIT IV: **13 Hours**
 CI/CD: Introduction to Continuous Integration, Continuous Delivery and Deployment, Benefits of CI/CD, Metrics to track CI/CD practices, Monitoring tools (Prometheus, Nagios, Grafana), Metrics: deployment frequency, lead time. **COs–CO4**

Self-Learning Topics: Automation & performance measurement

UNIT V: **12 Hours**
 Devops Maturity Model: Key factors of DevOps maturity model, stages of Devops maturity model, DevOps maturity Assessment. DevSecOps basics (security integration), Case studies of DevOps adoption **COs–CO5**

Self-Learning Topics: Automation & performance measurement

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries) **CO's/ PO's / PSO's**

1. From Traditional SDLC to Agile and DevOps Transformation- Limitations of Waterfall model in modern systems.
2. DevOps Architecture and Toolchain in Modern Software Development-
3. Implementing DevOps in Organizations: Challenges & Best Practices-Building Scalable CI/CD Pipelines and Monitoring Systems
4. DevOps Maturity Models and DevSecOps Integration

Text Books:

1. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations is considered the DevOps bible. It is written by Gene Kim, Jez Humble, Patrick Debois, and John Willis
2. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, is by Jez Humble and David Farley

Reference Books:

1. Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale by Jennifer Davis & Ryn Daniels.

Web References:

1. <https://www.udacity.com/course/intro-to-devops--ud611> - Good online course with sample exercises.
2. <http://www.edureka.co/devops> - Online Training covering high level process and tools. (Needs Registration)
3. https://www.edx.org/course?search_query=devops – Has no. of courses from MS and

[Redhat.](#)

4. [https://www.codementor.io/devops/tutorial - Basic Tutorial on DevOps.](https://www.codementor.io/devops/tutorial - Basic Tutorial on DevOps)

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define DevOps.
2. List the phases of the DevOps lifecycle.
3. What is Continuous Integration (CI)?
4. What is Continuous Deployment?
5. Name any two version control systems.
6. What is Agile methodology?
7. List any four DevOps tools.
8. What is Docker?
9. Define DevSecOps.
10. What is a DevOps maturity model?

L2: Understand

1. Explain the need for DevOps in software development.
2. Describe the principles of Agile methodology.
3. Explain the difference between CI and CD.
4. Summarize the DevOps lifecycle.
5. Describe the role of version control systems.
6. Explain Infrastructure as Code (IaC).
7. Discuss the importance of automation in DevOps.
8. Explain the DevOps pipeline.
9. Describe monitoring in DevOps.
10. Explain DevOps culture and collaboration.

L3: Apply

1. Apply Git commands to create and manage a repository.
2. Demonstrate how to commit and push code using Git.
3. Implement a simple CI pipeline using Jenkins.
4. Apply Docker to containerize an application.

5. Configure a basic CI/CD workflow.
6. Use a version control system for team collaboration.
7. Apply monitoring tools to track system performance.
8. Implement basic automation using scripts.
9. Use cloud services to deploy an application.
10. Apply DevOps practices in a sample project.

L4: Analysing

1. Analyze the differences between traditional SDLC and DevOps.
2. Examine the components of a DevOps pipeline.
3. Analyze the role of automation in DevOps efficiency.
4. Compare Agile and DevOps methodologies.
5. Analyze the working of CI/CD pipelines.
6. Examine bottlenecks in software deployment processes.
7. Analyze the impact of monitoring tools on performance.
8. Compare different version control systems.
9. Analyze challenges in DevOps adoption.
10. Examine the integration of DevSecOps in DevOps.

L5: Evaluating

1. Evaluate the effectiveness of DevOps in modern organizations.
2. Assess the advantages of CI/CD pipelines.
3. Evaluate different DevOps tools for a project.
4. Justify the use of automation in DevOps.
5. Evaluate the impact of DevOps on software quality.
6. Assess DevOps maturity in an organization.
7. Critique a given DevOps pipeline design.
8. Evaluate the role of cloud computing in DevOps.
9. Justify the need for DevSecOps.
10. Evaluate monitoring strategies in DevOps systems.

L6: Create

1. Design a complete DevOps pipeline for a web application.
2. Develop a CI/CD workflow using appropriate tools.
3. Create a DevOps architecture using cloud services.
4. Design a monitoring system for an application.
5. Develop a containerized application using Docker.
6. Create an automation script for deployment.
7. Design a DevOps strategy for a startup company.
8. Develop a version control workflow for a team project.
9. Create a DevSecOps pipeline integrating security practices.
10. Design a DevOps maturity improvement plan.

**Chairperson
Board of Studies (CSE)**

R24CSOE06

COMPUTER NETWORKS

3 0 0 3

(Open Elective-II)

Course Objectives:

1. **To understand** the fundamental concepts of computer networks, including network models (OSI and TCP/IP) and different types of networks (LAN, MAN, WAN).
2. **To analyze** the functions and protocols of various network layers such as Physical, Data Link, Network, Transport, and Application layers.
3. **To develop** the ability to evaluate networking mechanisms like routing, congestion control, error control, and reliable data transmission in modern communication systems.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs														Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2		
R24CSOE06.1	Explain the basic concepts of computer networks, network types, reference models, and transmission media.	3	2	-	-	1	-	-	-	-	-	-	2	1	L1, L2	
R24CSOE06.2	Analyze data link layer protocols, error detection and correction techniques, flow control, and multiple access methods.	3	3	1	-	2	-	-	-	-	-	-	2	2	L2, L3	
R24CSOE06.3	Evaluate routing algorithms, packet switching techniques, and congestion control mechanisms in network layer.	3	3	2	1	2	1	-	-	-	-	-	3	2	L3, L4	
R24CSOE06.4	Understand transport layer protocols (TCP/UDP), connection management, and reliability mechanisms.	3	2	2	-	2	-	-	-	-	-	-	3	2	L4, L5	
R24CSOE06.5	Describe application layer protocols such as HTTP, DNS, and web architecture.	2	1	-	-	2	-	-	-	1	-	1	2	3	L5, L6	

SYLLABUS**UNIT I: Introduction & Physical Layer****15 Hours**

Introduction: Network Types: LAN, MAN, WAN. Network Reference models: The OSI Reference Model, The TCP/IP Reference Model. Software Defined Networking – Introduction

Physical Layer: Introduction to Guided Transmission Media- Twisted-pair cable, Coaxial cable and Fiber optic cable. Unguided media: Wireless: Radio waves, microwaves, infrared.

Physical Layer Impairments & Signal Integrity –Basics

CO's-CO1

Self-Learning Topics: Digital Modulation and Multiplexing.

UNIT II: Data Link Layer and Media Access Control

15 Hours

Design Issues: Services Provided to the Network Layer

Error Detection and Correction: Error Detection: CRC, Checksum.

Error Correction: Framing -Bit and Byte stuffing; Adaptive Frame Size Mechanisms – Basics

Flow Control: Sliding Window Protocols. Stop and wait ARQ; Go Back N, Selective repeat

Channel allocation methods: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, Machine Learning-Based Access Control (in Random Access Networks) – Introduction

Controlled Access: Token Bus, Token Ring. Bridges and Virtual LANs, Energy-Efficient Ethernet (IEEE 802.3az) - Introduction **CO's–CO2**

Self-Learning Topics: IEEE 802, 802.11 and 802.16. Data link layer switching,

UNIT III: Network Layer and Congestion Control

12 Hours

Design Issues: Services provided to the Transport layer, Store and Forward Packet Switching, Implementation of Connectionless Service (Datagram Networks), Implementation of Connection Oriented Service (Virtual Circuits), Comparison of Virtual Circuit and Datagram Networks, Hybrid Networking Models Overlay Networks – Basics

Routing Algorithms: The Optimality principle, shortest path, Flooding, Flow based Distance vector, Link state, Hierarchical, Broadcast routing, Multicast & Event Distribution Models – Introduction

Congestion Control algorithms: General principles of congestion control, Congestion prevention policies, choke packets, Load shedding, and Jitter Control, Modern Choke Mechanism – Introduction **CO's–CO3**

Self-Learning Topics: Internetworking, Fragmentation, Network layer in the internet

UNIT IV: Transport Layer

15 Hours

The Transport Layer: Elements of transport protocols – addressing, establishing a connection, releasing connection, flow control and buffering and crash recovery, UDP, QUIC Protocol – Basics

The Internet Transport Protocol: TCP- reliable Byte Stream (TCP) end to end format, segment format, connection establishment and termination, sliding window revisited, adaptive retransmission, Connection Pooling and Keep-Alive Mechanisms – Introduction **COs–CO4**

Self-Learning Topics: OSPF, BGP, Internet multicasting, Mobile IP, IPV6.ICMP-ARP- DHCP

UNIT V: Application Layer

13 Hours

WWW: Architecture- Client (Browser), Server, Uniform Resource Locator, **Statelessness & Session Management Techniques – Introduction (New)**

HTTP: HTTP Transaction, HTTP Operational Model and Client/Server Communication, HTTP Generic Message Format, HTTP Request Message Format, HTTP Response Message Format, Error Handling & Status Code Design – Basics

The Domain Name System: The DNS Name Space, Resource Records, Name Servers, DNSSEC & Security Mechanisms – Introduction **CO's–CO5**

Self-Learning Topics: Electronic Mail

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. **Computer Networks in Cloud Computing** (*Cloud networking in platforms like AWS, Azure, GCP; Virtual networks, load balancing, network security, SDN concepts*) CO3–CO5, PO1, PO2, PO3, PO5
2. **Network Security and Cyber security Practices** (*Firewalls, intrusion detection systems, secure protocols, network attacks and prevention techniques*) CO2, CO3, CO5, PO1, PO2, PO3, PO5, PO11
3. **Emerging Networking Technologies** (*IoT networking, 5G communication, network virtualization, software-defined networking (SDN)*) CO1, CO3, CO5, PO1, PO2, PO3, PO5, PO11

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. **Dr. Sanjay Sharma**, *Computer Networks*, 2nd Edition, Katson Books, New Delhi, 2019
2. **B. A. Forouzan**, *Data Communications and Networking*, 5th Edition, McGraw Hill Education (India), 2017.
3. *Computer Networks* – Andrew S Tanenbaum, Fifth Edition. Pearson Education / PHI
4. *Computer Networks*, 5ed, David Patterson, Elsevier

Reference Books:

1. *Data Communications and Networks* – Achut S Godbole, Atul Kahate
2. *Computer Networks*, Mayank Dave, CENGAGE
3. **N. N. Sakhare and A. V. Dhumane**, *Computer Network*, 1st Edition, Technical Publications, Pune, 2018.
4. **D. S. Gaikwad and N. D. Lambe**, *Computer Network Technology*, 1st Edition, Technical Publications, Pune, 2016.
5. **Achyut S. Godbole**, *Data Communications and Networks*, 2nd Edition, McGraw Hill Education (India), 2013.
6. *An Engineering Approach to Computer Networks*-S. Keshav, 2nd Edition, Pearson Education

Web References:

1. NPTEL (SWAYAM): <https://nptel.ac.in>
2. Geeks for Geeks: <https://www.geeksforgeeks.org>
3. Cisco Networking Academy: <https://www.netacad.com>
4. Tutorials Point: <https://www.tutorialspoint.com>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is LAN, MAN, and WAN? (Unit I)
2. Define OSI reference model. (Unit I)
3. What is TCP/IP model? (Unit I)
4. List guided and unguided transmission media. (Unit I)
5. Define CRC and checksum. (Unit II)
6. What is framing in data link layer? (Unit II)
7. Define CSMA/CD and CSMA/CA. (Unit II)
8. What is Token Ring? (Unit II)
9. Define routing and packet switching. (Unit III)
10. What is congestion control? (Unit III)
11. Define TCP and UDP. (Unit IV)
12. What is HTTP and DNS? (Unit V)
13. Define Software-Defined Networking (SDN).
14. What is the main goal of using machine learning in channel access control?
15. What is meant by connection migration in QUIC?

L2: Understand

1. Explain OSI vs TCP/IP models. (Unit I)
2. Describe twisted pair, coaxial, and fiber optic cables. (Unit I)
3. Explain bit stuffing and byte stuffing. (Unit II)
4. Describe error detection and correction techniques. (Unit II)
5. Explain sliding window protocols. (Unit II)
6. Describe distance vector and link state routing. (Unit III)
7. Explain congestion control policies and jitter control. (Unit III)
8. Describe TCP connection establishment and termination. (Unit IV)
9. Explain flow control and buffering in transport layer. (Unit IV)
10. Explain HTTP message format. (Unit V)
11. Describe DNS name space and resource records. (Unit V)
12. How do physical layer impairments influence network performance metrics like throughput and latency?
13. Explain how DNS cache poisoning attacks occur in computer networks.

L3: Apply

1. Apply OSI model to analyze a real-world network. (Unit I)
2. Demonstrate CRC calculation for error detection. (Unit II)
3. Apply Stop-and-Wait ARQ in a transmission scenario. (Unit II)
4. Apply Go-Back-N or Selective Repeat protocol. (Unit II)
5. Solve routing using shortest path algorithm. (Unit III)
6. Apply distance vector algorithm to a network topology. (Unit III)
7. Demonstrate congestion control using choke packets. (Unit III)

8. Apply TCP sliding window mechanism. (Unit IV)
9. Demonstrate TCP three-way handshake. (Unit IV)
10. Apply HTTP protocol in client-server communication. (Unit V)
11. A wireless network shows frequent retransmissions due to noise. Apply adaptive frame sizing to reduce retransmission overhead. Explain the steps involved.
12. A service is temporarily overloaded. Apply correct status code and explain how graceful degradation can be implemented

L4: Analysing

1. Analyze OSI and TCP/IP models. (Unit I)
2. Compare guided vs unguided transmission media. (Unit I)
3. Analyze CSMA/CD vs CSMA/CA protocols. (Unit II)
4. Compare Stop-and-Wait, Go-Back-N, and Selective Repeat. (Unit II)
5. Analyze performance of link state vs distance vector routing. (Unit III)
6. Examine congestion control techniques and load shedding. (Unit III)
7. Analyze TCP vs UDP performance. (Unit IV)
8. Compare TCP retransmission strategies. (Unit IV)
9. Analyze HTTP working in web communication. (Unit V)
10. Compare DNS with other naming mechanisms. (Unit V)
11. Compare and analyze the trade-off between energy efficiency and performance in Energy-Efficient Ethernet under varying traffic conditions.
12. Compare and analyze hybrid networking models (combining underlay and overlay) in terms of control, scalability, and complexity.

L5: Evaluating

1. Evaluate the efficiency of OSI model in modern networks. *(Unit I)*
2. Assess advantages and disadvantages of different transmission media. *(Unit I)*
3. Evaluate error detection vs error correction techniques. *(Unit II)*
4. Critically evaluate different MAC protocols. *(Unit II)*
5. Evaluate effectiveness of congestion control algorithms. *(Unit III)*
6. Assess routing algorithms for large-scale networks. *(Unit III)*
7. Evaluate TCP congestion control mechanisms. *(Unit IV)*
8. Assess performance of UDP in real-time applications. *(Unit IV)*
9. Evaluate HTTP protocol efficiency in web applications. *(Unit V)*
10. Assess DNS scalability and reliability. *(Unit V)*
11. Evaluate how event ordering and consistency requirements affect the design of multicast or event-driven architectures.
12. Evaluate the suitability of adaptive choke mechanisms in handling burst traffic in high-traffic web applications.
13. Evaluate whether connection pooling or keep-alive is more effective for microservices communication. Justify.

L6: Create

1. Design a network using OSI model concepts. *(Unit I)*
2. Develop a framing technique for efficient data transmission. *(Unit II)*
3. Design a MAC protocol for a given network scenario. *(Unit II)*
4. Develop a routing algorithm for optimized path selection. *(Unit III)*
5. Design a congestion control mechanism for high-traffic networks. *(Unit III)*

6. Create a TCP-based communication model for reliable transfer. (*Unit IV*)
7. Design an efficient transport layer protocol for real-time systems. (*Unit IV*)
8. Develop a client-server application using HTTP protocol. (*Unit V*)
9. Design a DNS system for a distributed network. (*Unit V*)
10. Create a secure application layer protocol for data exchange. (*Unit V*)

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To understand the fundamental concepts, architecture, and paradigms of mobile computing and GSM.
2. To comprehend the motivation for specialized MAC techniques in wireless communications, addressing issues such as hidden and exposed terminals, near and far terminals and IEEE 802.11.
3. To analyze the IP and Mobile IP network layers, focusing on packet delivery, handover management, location management, registration, tunneling, encapsulation and DHCP.
4. To understand conventional TCP/IP protocols and specialized protocols like Indirect TCP, Snooping TCP, and Mobile TCP for mobile networks and database issues in mobile computing
5. To introduce the concept, applications, and challenges of Mobile Ad hoc Networks (MANETs), and to explore various routing algorithms such as DSR, AODV, and DSDV.

Course Outcomes:

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSOE 07.1	Understand fundamentals of mobile communication systems, GSM architecture, services, and evolution from 1G to 5G, including QoS, security, and mobile device limitations.	3	2	1	-	2	1	-	-	-	-	1	2	1	L1, L2
R24CSOE 07.2	Analyze wireless MAC protocols and understand multiple access techniques, IEEE 802.11 standards, and energy-efficient MAC mechanisms.	3	3	2	1	2	-	-	-	-	-	1	2	2	L2, L3
R24CSOE 07.3	Explain mobile network layer concepts including Mobile IP, routing, tunneling, IPv6, DHCP, and mobility management.	3	3	2	2	2	1	-	-	-	-	1	3	2	L3, L4
R24CSOE 07.4	Evaluate transport layer protocols in mobile environments and analyze performance issues, congestion control, and enhancements like Indirect TCP and Mobile TCP.	3	2	2	2	3	-	-	-	-	-	1	3	2	L4, L5

R24CSOE 07.5	Understand MANETs, routing protocols, mobile ad hoc challenges, IoT integration, and mobile computing platforms and security mechanisms.	3	3	2	2	2	2	1	1	1	1	2	3	3	L5, L6
-----------------	--	---	---	---	---	---	---	---	---	---	---	---	---	---	-----------

SYLLABUS

UNIT-I: Introduction

15 Hours

Mobile Communications, Mobile Computing – Paradigm, Promises/Novel Applications and Impediments and Architecture, Mobile and Handheld Devices, Limitations of Mobile and Handheld Devices, GSM – Services, System Architecture, Radio Interfaces, Protocols, Localization and Calling, Handover, Security, New Data Services, Quality of Service (QoS) in Mobile Networks, Evolution of Mobile Communication Systems (1G to 5G).

CO's-CO1

Self-Learning Topics: 5G and Beyond (6G concepts, future trends in mobile communication)

Case Studies of Mobile Computing Applications (Smart Cities, Mobile Banking, Telemedicine)
Comparison of GSM with LTE and 5G architectures.

UNIT-II: Wireless Medium Access Control (MAC)

13 Hours

Motivation for a specialized MAC (Hidden and Exposed Terminals, Near and Far Terminals), SDMA, FDMA, TDMA, CDMA, MAC Protocols for Wireless LANs (IEEE 802.11, CSMA/CA, RTS/CTS), Energy-Efficient MAC Protocols.

CO's-CO2

Self-Learning Topics: Advanced IEEE 802.11 Standards (802.11ac, 802.11ax – Wi-Fi 6), MAC Layer Security Issues and Solutions, Cognitive Radio and Dynamic Spectrum Access.

UNIT-III: Mobile Network Layer

13 Hours

IP and Mobile IP Network Layers, Packet Delivery and Handover Management, Location Management, Registration, Tunnelling and Encapsulation, Route Optimization, DHCP, IPv6 in Mobile Networks, Network Security in Mobile IP.

CO's-CO3

Self-Learning Topics: Software Defined Networking (SDN) in Mobile Networks, Network Function Virtualization (NFV), Mobility Management in 5G Networks

UNIT-IV: Mobile Transport Layer

14 Hours

Conventional TCP/IP Protocols, Indirect TCP, Snooping TCP, Mobile TCP, Other Transport Layer Protocols for Mobile Networks, Congestion Control in Mobile Networks, Performance Issues of TCP in Wireless Networks.

CO's-CO4

Self-Learning Topics: QUIC Protocol and HTTP/3, Cross-layer Optimization in Mobile Networks, SCTP (Stream Control Transmission Protocol), Impact of Wireless Errors on Transport Protocols

UNIT-V: Mobile Ad hoc Networks (MANETs)

15 Hours

Introduction, Applications & Challenges of a MANET, Routing, Classification of Routing Algorithms, Algorithms such as DSR, AODV, DSDV, Mobile Agents, Service Discovery, Protocols and Platforms for Mobile Computing: WAP, Bluetooth, XML, J2ME, Java Card, PalmOS, Windows CE, SymbianOS, Linux for Mobile Devices, Android, Security in MANETs (Attacks and Defence Mechanisms), Internet of Things (IoT) in Mobile Computing.

CO's-CO5

Self-Learning Topics: Vehicular Ad hoc Networks (VANETs), Delay Tolerant Networks (DTNs), Block chain in Mobile Ad hoc Networks, Edge Computing and Fog Computing in Mobile Environments.

Board of Studies : Computer Science and Engineering

Approved in BoS No: 03

Approved in ACM No: 03

Text Books

1. Mobile Communications, Addison-Wesley, 2nd Edition, 2009.
2. Mobile Computing, Oxford University Press, 1st Edition, 2007.
3. Wireless Communications: Principles and Practice, Pearson Education, 2nd Edition, 2002.
4. Computer Networking: A Top-Down Approach, Pearson, Latest Edition.
5. Wireless and Mobile Networks, Pearson Education, 2nd Edition.

Reference Books:

1. **Jochen Schiller**
Mobile Communications, Pearson Education
(Comprehensive book covering GSM, wireless systems, and mobile networking concepts)
2. **Raj Kamal**
Mobile Computing, Oxford University Press
(Covers mobile architecture, protocols, and applications in detail)
3. **William Stallings**
Wireless Communications and Networks, Pearson Education
(Strong foundation for MAC, routing, and wireless protocols)
4. **Asoke K. Talukder & Roopa R. Yavagal**
Mobile Computing Tata McGraw Hill
(Good for Mobile IP, transport layer, and application-level concepts)
5. **Sandeep Singhal et al.**
The Wireless Application Protocol (WAP)
Addison-Wesley
(Useful for mobile application protocols and early mobile platforms)

Web References:

1. <https://www.3gpp.org>
(Official standards for GSM, LTE, and 5G technologies)
2. <https://www.cisco.com>
(Networking concepts, wireless technologies, and mobility solutions)
3. <https://www.geeksforgeeks.org/mobile-computing/>
(Simple tutorials and explanations of mobile computing concepts)
4. https://www.tutorialspoint.com/mobile_computing/
(Structured notes on mobile communication and networking)
5. <https://www.ieee.org>
(Research papers on MANETs, MAC protocols, and wireless networks)
6. <https://nptel.ac.in>
(Free video lectures on Mobile Computing and Wireless Networks)

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--

L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1 (Remember)

1. Define Mobile Computing?
2. What is a mobile communication system?
3. List the services provided by GSM?
4. What is handover in GSM?
5. Define QoS (Quality of Service) in mobile networks?
6. What is FDMA, TDMA, and CDMA?
7. Define hidden terminal problem?
8. What is Mobile IP?
9. Define tunneling in Mobile IP?
10. What is MANET?

L2 (Understand)

1. Explain the architecture of a mobile computing system?
2. Describe limitations of mobile and handheld devices?
3. Explain GSM system architecture and its components?
4. Describe the evolution from 1G to 5G mobile systems?
5. Explain wireless MAC issues like hidden and exposed terminals?
6. Describe IEEE 802.11 MAC protocol and CSMA/CA?
7. Explain Mobile IP and its working?
8. Describe packet delivery and handover management?
9. Explain TCP issues in wireless/mobile networks?
10. Describe MANET characteristics and challenges?

L3 (Apply)

1. Illustrate GSM call setup with a diagram?
2. Apply TDMA/FDMA concepts to allocate channels?
3. Demonstrate RTS/CTS mechanism to avoid collisions?
4. Apply Mobile IP to route packets to a mobile node?
5. Solve a handover scenario in cellular networks?
6. Demonstrate congestion control in a wireless network?
7. Apply AODV routing in a small MANET example?
8. Demonstrate DSR routing with route discovery?
9. Apply DHCP for IP allocation in mobile networks?
10. Illustrate Bluetooth communication setup?

L4 (Analyze)

1. Compare GSM, LTE, and 5G architectures?
2. Analyze the impact of mobility on network performance?
3. Compare FDMA, TDMA, CDMA, and SDMA?

4. Analyze MAC layer issues in wireless networks?
5. Compare Mobile IP with traditional IP?
6. Analyze TCP performance issues in wireless networks?
7. Compare routing protocols: DSR, AODV, and DSDV?
8. Analyze security threats in MANETs?
9. Compare IPv4 and IPv6 in mobile environments?
10. Analyze QoS challenges in mobile networks?

L5 (Evaluate)

1. Evaluate the advantages and limitations of GSM?
2. Justify the need for specialized MAC protocols in wireless networks?
3. Evaluate different multiple access techniques?
4. Assess Mobile IP efficiency in real-world scenarios?
5. Evaluate TCP variants for mobile networks (Indirect TCP, Snooping TCP)?
6. Assess routing protocols used in MANETs?
7. Evaluate security mechanisms in mobile networks?
8. Assess the performance of IEEE 802.11 protocols?
9. Evaluate energy-efficient MAC protocols?
10. Assess IoT integration with mobile computing systems?

L6 (Create)

1. Design a mobile computing architecture for a smart city?
2. Propose improvements for GSM or 5G networks?
3. Design a MAC protocol for energy-efficient communication?
4. Develop a routing protocol for MANETs?
5. Design a handover mechanism for seamless mobility?
6. Create a congestion control method for mobile networks?
7. Design a secure Mobile IP framework?
8. Develop an application using mobile computing (e.g., telemedicine)?
9. Propose a QoS model for 5G networks?
10. Design an IoT-based mobile computing system?

**Chairperson
Board of Studies (CSE)**

R24CSOE08

Software Project Management
(Open Elective-II)

3 0 0 3

Course Objectives:

1. To understand conventional software management and software economics, and apply cost estimation and improvement strategies to enhance software productivity and project performance.
2. To understand the transition from conventional to modern software management principles and apply iterative development practices across software life cycle phases.
3. To understand model-based software architectures and apply workflow-driven iterative planning techniques for effective cost, schedule, and project management.
4. To understand project organization structures and apply metrics, process control techniques, and COCOMO-based cost estimation for effective software project management.
5. To apply effort estimation, scheduling, quality planning, and risk management techniques for effective and reliable software project execution.

Course Outcomes:

At the end of the course, students will be able to

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSOE 08.1	Estimate overall cost of a software project.	3	2	1	–	2	–	–	–	–	3	1	–	1	L1, L2
R24CSOE 08.2	Explain software development process.	3	2	2	–	2	–	–	–	–	3	1	1	–	L2, L3
R24CS.OE 08.3	Distinguish workflows of process.	3	2	2	1	3	–	–	1	–	3	1	1	1	L3, L4
R24CSOE 08.4	Design project organization structure & analyze quality.	3	2	1	2	3	–	–	1	–	3	1	–	1	L4, L5
R24CSOE 08.5	Estimate effort and schedule needed for project.	3	3	2	2	2	1	1	2	1	–	–	1	–	L5, L6

SYLLABUS**UNIT I:****13 Hours****Conventional Software Management:** The Waterfall Model, Conventional Software Management Performance.**Evolution of Software Economics:** Software Economics, Pragmatic Software Cost Estimation.**Improving Software Economics:** Reducing Software Product Size, Improving Software Processes, Improving Team Effectiveness, Improving Automation through Software Economics.**CO's–CO1****Self-Learning Topics:** Limitations of Waterfall model, Process improvement models**UNIT II:****14 Hours****The Old Way and the New:** The Principles of Conventional Software Engineering, The Principles

of Modern Software Management, Transitioning to an Iterative Process.

Life Cycle Phases: Engineering and Production Stages, Inception Phase, Elaboration Phase, Construction Phase, Transition Phase. **CO's–CO2**

Self-Learning Topics: Limitations of conventional software engineering principles, Differences between Waterfall and Iterative processes

UNIT III:

13 Hours

Model Based Software Architectures: A Management Perspective, A Technical Perspective.

Workflows of the Process: Software Process Workflows, Iteration Workflows.

Iterative Process Planning: Work Breakdown Structures, Planning Guidelines, The Cost and Schedule Estimating Process, The Iteration Planning Process. **CO's–CO3**

Self-Learning Topics: Basics of model-based software architecture (UML), Cost and schedule estimation techniques

UNIT IV:

15 Hours

Project Organization and Responsibilities: Line-Of-Business Organizations, Project Organizations, Evolution of Organizations.

Project Control and Process Instrumentation: The Seven Core Metrics, Management Indicators, Quality Indicators Modern Project Profiles. The COCOMO Cost Estimation Model: COCOMO. **COs–CO4**

Self-Learning Topics: Basics of project control and monitoring, Fundamentals of COCOMO model

UNIT V:

12 Hours

Effort Estimation and Scheduling: Effort Estimation, Scheduling.

Quality Planning: Quality Concepts, Quantitative Quality Management Planning.

RISK MANAGEMENT: Risk Assessment, Risk Control.

CO's-CO5

Self-Learning Topics: Quality planning techniques, Risk identification methods

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Textbooks:

1. Walker Royce, “Software Project Management – A UnifiedFramework”, 1stEdition, Pearson Education, 2002.
2. Pankaj Jalote, “Software Project Management in Practice”, 1stEdition, Pearson Education, 2005.
3. Software Project Management, Bob Hughes & Mike Cotterell, TATA McGraw-Hill.

Reference Books:

1. Bob Hughes, “Mike Cotterell, Rajib Mall, Software Project Management”, 5thEdition, McGraw-Hill Higher Education, 2011.
2. Joel Henry, “Software Project Management”, 1st Edition, Pearson Education, 2006.
3. Norman E. Fenton, Shari Lawrence Pfleeger, “Software Metrics: A Rigorous and Practical Approach “, 1st Edition, PWS Publishing Company, 1997

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define the Waterfall model and list its phases.
2. What is software economics? State its key components.
3. List the principles of conventional software engineering and name the life cycle phases.
4. Define iterative process and identify its key stages.
5. Define model-based software architecture and list its key elements.
6. What is a Work Breakdown Structure (WBS)? List its components.
7. Define project organization and list types of software organizational structures.
8. What is the COCOMO model? State its basic concepts.
9. Define effort estimation and scheduling in software project management.
10. What are software quality concepts? List key quality attributes.

L2: Understand

1. Explain the concept of conventional software management performance.
2. Describe the importance of pragmatic software cost estimation in project management.
3. Explain the principles of modern software management.
4. Describe the activities carried out in the Inception and Elaboration phases.
5. Explain the management and technical perspectives of model-based software architectures.
6. Describe different types of software process workflows and iteration workflows.
7. Explain the roles and responsibilities in a project organization.
8. Describe the seven-core metrics used in software project control.
9. Explain the process of quantitative quality management planning.
10. Describe risk assessment and risk control in software projects.

L3: Apply

1. Apply software cost estimation techniques to estimate effort for a small project. (L3)
2. Illustrate how reducing software product size improves software economics. (L3)
3. Apply the concept of iterative development to manage a small software project. (L3)
4. Illustrate the transition from conventional to iterative processes with a suitable example. (L3)
5. Apply WBS to decompose a software project into manageable tasks. (L3)
6. Illustrate the process of cost and schedule estimation for a software project. (L3)
7. Apply the COCOMO model to estimate effort for a software project. (L3)
8. Illustrate how project control and monitoring techniques are used in managing software projects. (L3)

9. Apply scheduling techniques to prepare a simple project schedule. (L3)
10. Illustrate quality planning techniques for a software development project. (L3)

L4: Analysing

1. Analyze the limitations of the Waterfall model in modern software development.
2. Compare different approaches to improving software processes and their impact on productivity.
3. Analyze the limitations of conventional software engineering principles in current software development practices.
4. Differentiate between Waterfall and Iterative processes with respect to flexibility, risk, and delivery.
5. Analyze the role of iteration workflows in improving software development efficiency.
6. Compare different cost and schedule estimation techniques used in software projects.
7. Analyze the evolution of software organizations and their impact on project management.
8. Differentiate between management indicators and quality indicators in project control.
9. Analyze different risk identification methods used in software projects.
10. Compare various effort estimation techniques and their suitability.

L5: Evaluating

1. Evaluate the effectiveness of team improvement strategies in enhancing software project performance.
2. Assess the role of process improvement models in optimizing software development outcomes.
3. Evaluate the effectiveness of iterative processes over traditional approaches in handling changing requirements.
4. Assess the importance of life cycle phases in ensuring successful software development.
5. Evaluate the effectiveness of model-based architectures in managing complex software systems.
6. Assess the importance of planning guidelines in iterative process planning.
7. Evaluate the effectiveness of project metrics in controlling software development.
8. Assess the role of COCOMO in improving software cost estimation accuracy.
9. Evaluate the effectiveness of quality planning in ensuring software reliability.
10. Assess the role of risk management in successful project execution.

L6: Create

1. Propose a strategy to improve software economics using automation and process optimization.
2. Design a framework to enhance team effectiveness and reduce development cost in a software project.
3. Design an iterative development plan covering all life cycle phases for a software project.
4. Propose a strategy to transition an organization from a conventional to a modern software management approach.
5. Design an iteration plan including WBS, cost estimation, and scheduling for a software project.
6. Propose a workflow-driven approach for managing a software development process using model-based architecture.
7. Design a project control framework using core metrics and indicators.
8. Propose an organizational structure suitable for managing a large-scale software project.

9. Design a risk management plan including risk assessment and control strategies.
10. Propose a comprehensive plan integrating effort estimation, scheduling, and quality management for a software project.

**Chairperson
Board of Studies (CSE)**

R24CSOE09**Object Oriented through JAVA****3 0 0 3**

(Computer Science and Engineering)

Course Objectives:

1. Identify Java language components and how they work together in applications
2. Learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
3. Learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
4. Understand how to design applications with threads in Java
5. Understand how to use Java APIs for program development

Course Outcomes

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSOE 09.1	Analyze problems, design solutions using OOP principles, and implement them efficiently in Java	3	2	1	-	2	-	-	-	-	-	1	1	-	L1, L2
R24CSOE 09.2	Design and implement classes to model real-world entities, with a focus on attributes, behaviors, and relationships between objects	3	2	2	-	3	-	-	1	1	-	2	1	-	L2, L3
R24CSOE 09.3	Demonstrate an understanding of inheritance hierarchies and polymorphic behaviors, including method overriding and dynamic method dispatch.	3	3	2	1	3	-	1	1	1	-	2	0	-	L3, L4
R24CSOE 09.4	Apply Competence in handling exceptions and errors to write robust and fault-tolerant code.	3	2	3	2	3	-	-	2	1	1	2	-	1	L4, L5
R24CSOE 09.5	Perform file input/output operations, including reading from and writing to files using Java I/O classes, graphical user interface (GUI) programming using JavaFX.	3	2	2	1	3	-	-	1	1	-	2	-	1	L5, L6

SYLLABUS**UNIT I****15 Hours****Object Oriented Programming:** Basic concepts, Principles, Program Structure in Java:

Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments.

Data Types, Variables, and Operators : Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf () Method.

Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (--) Operators, Ternary Operator, Relational Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator? Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, for–Each for Loop, Break Statement. **CO's–CO1**

Self-Learning Topics: Programming Style, Static Variables and Methods, Attribute Final.

UNIT II

15 Hours

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Constructor Methods for Class, Overloaded Constructor Methods, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods. **CO's – CO2**

Self-Learning Topics: Accessing Private Members of Class, Nested Classes.

UNIT III

13 Hours

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Two-dimensional Arrays.

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Static Methods in Interface, Functional Interfaces. **CO's – CO3**

Self-Learning Topics: Arrays of Varying Lengths, Default Methods in Interfaces.

UNIT IV

15 Hours

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Access Control, Packages in Java SE, Java. lang Package and its Classes, Class Object, Enumeration, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Class Instant (java. time. Instant).

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.

Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java(Text Book 2) **CO's – CO4**

Self-Learning Topics: Path and Class Path, class Math.

UNIT V**14 Hours**

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching;

Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.

Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and Image, event handling, laying out nodes in scene graph. (Text Book 3)

CO's – CO5

Self-Learning Topics: Class String Buffer, Thread Priority-Synchronization.

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
2. Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
3. JAVA for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

References Books:

1. The complete Reference Java, 11th edition, Herbert Schildt, TMH
2. Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

Online Resources:

1. <https://nptel.ac.in/courses/106/105/106105191/>
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define Object-Oriented Programming. Explain its basic concepts.
2. List and explain different data types available in Java.
3. Define a class and an object in Java. Explain their basic features.
4. What are constructors? List different types of constructors in Java.

5. Define an array. List different types of arrays in Java.
6. What is inheritance? List various types of inheritance supported in Java.
7. Define a package in Java. List its advantages.
8. What are exceptions? Classify different types of exceptions in Java.
9. Define a String in Java. List important methods of the String class.
10. What is multithreading? List advantages of multithreaded programming.
11. Define a thread. List different states of a thread in Java.

L2: Understand

1. Explain the structure of a simple Java program with an example.
2. Differentiate between variables and constants in Java with examples.
3. Describe constructor overloading with suitable examples.
4. Explain the concept of passing arguments by value and by reference in Java.
5. Describe the process of inheritance with a suitable example.
6. Explain the role of the super keyword in inheritance.
7. Explain how to define and import packages in Java with examples.
8. Describe the hierarchy of exception classes in Java.
9. Explain the use of try, catch, and finally blocks with an example.
10. Explain the concept of immutable strings in Java.
11. Describe the life cycle of a thread with a diagram.

L3: Apply

1. Write a Java program demonstrating the use of command line arguments.
2. Write a Java program to illustrate the use of arithmetic and relational operators.
3. Write a Java program to demonstrate the creation of a class and objects.
4. Develop a Java program to illustrate constructor overloading.
5. Write a Java program to implement an interface and demonstrate its usage.
6. Implement a Java program to illustrate dynamic method dispatch.
7. Write a Java program to create and use a package.
8. Develop a Java program demonstrating exception handling using multiple catch blocks.
9. Write a Java program to read input using Scanner class and display output.
10. Write a Java program demonstrating string comparison and modification methods.
11. Develop a Java program to create and run multiple threads.

L4: Analysing

1. Compare switch statement with if–else statement in Java.
2. Analyze operator precedence and associativity with suitable examples.
3. Differentiate between method overloading and method overriding with examples.
4. Analyze the role of access control modifiers in ensuring data security.
5. Analyze different types of inheritance with suitable diagrams.
6. Compare abstract classes and interfaces in Java.
7. Differentiate between checked and unchecked exceptions with examples.
8. Analyze the difference between throw and throws keywords.
9. Compare byte streams and character streams in Java I/O.
10. Differentiate between String, StringBuffer, and StringBuilder.
11. Analyze the need for multithreading in multi-core processors.

L5: Evaluating

1. Justify the use of symbolic constants over literal constants.
2. Assess the advantages and limitations of using ternary operator over if–else.
3. Evaluate the importance of using constructors in object initialization.
4. Justify the use of final keyword in classes and methods.

5. Justify the use of inheritance for code reusability and extensibility.
6. Assess the role of interfaces in achieving multiple inheritance in Java.
7. Evaluate the importance of exception handling in Java applications.
8. Justify the use of packages for organizing large Java programs.
9. Assess the advantages of using Java I/O streams for file handling.
10. Evaluate the advantages and limitations of immutable strings.
11. Justify the need for synchronization in multithreaded applications.

L6: Create

1. Develop a Java program that demonstrates all types of loops and decision-making statements.
2. Create a Java program to accept user input and display formatted output using printf().
3. Design a Java class with multiple constructors and demonstrate constructor overloading.
4. Develop a Java program to illustrate method overriding using inheritance.
5. Develop a Java application demonstrating multilevel inheritance and method overriding.
6. Create a Java program that implements multiple interfaces.
7. Create a Java program to read and write data to a file using character streams.
8. Design a real-world application integrating packages, exception handling, and file operations.
9. Develop a multithreaded Java application demonstrating synchronization.
10. Create a JavaFX GUI application with event handling and layout management.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. Understand different types of computing (parallel, cloud, etc.).
2. Learn basics of cloud computing and why it is needed.
3. Know cloud architecture and how it works.
4. Study different cloud services (IaaS, PaaS, SaaS).
5. Get idea about cloud providers and new technologies.

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSO E10	Understand the concepts of computing paradigms.	3	2	1	-	2	-	-	-	-	-	-	3	1	L1, L2
R24CSO E10	Understand cloud computing concepts and models.	3	2	-	-	2	-	-	-	-	-	-	3	2	L2, L3
R24CSO E10	Describe cloud architecture and migration.	3	3	2	-	3	-	-	-	-	-	-	3	3	L3, L4
R24CSO E10	Understanding cloud service providers.	3	2	-	-	3	-	-	-	-	2	-	3	2	L4, L5
R24CSO E10	Know basic use of platforms like Amazon Web Services, Google Cloud Platform, and Microsoft Azure.	3	1	-	-	3	-	-	-	-	-	2	2	3	L5, L6

SYLLABUS**UNIT-I****12 Hours**

Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Bio computing, Mobile Computing, Quantum Computing, Optical Computing, Nano computing. Introduction to hypervisors, Simple tools: VMware Microsoft Hyper-V

CO's- CO1

Self-Learning Topics: Difference between parallel and distributed computing

UNIT-II**12 Hours**

Cloud Computing Fundamentals: Motivation for Cloud Computing, The Need for Cloud Computing, Defining Cloud Computing, Definition of Cloud computing, Cloud Computing Is a Service, Cloud Computing Is a Platform, Principles of Cloud computing, Five Essential Characteristics, Deployment Model Basic Concepts: Scalability, Fault tolerance, Introduction to cloud security

Self-Learning Topics: Micro services vs Monolithic architecture (basic idea) **CO's- CO2**
UNIT -III **15 Hours**

Cloud Computing Architecture and Management: Cloud architecture, Layer, Anatomy of the Cloud, Network Connectivity in Cloud Computing, Applications, on the Cloud, Managing the Cloud, Managing the Cloud Infrastructure Managing the Cloud application, Migrating Application to Cloud, Phases of Cloud Migration Approaches for Cloud Migration, Introduction to SDK.

Self-Learning Topics: Simple use of Cloud SDK tools **CO's- CO3**

UNIT-IV **15 Hours**

Cloud Service Models: Infrastructure as a Service, Characteristics of IaaS. Suitability of IaaS, Pros and Cons of IaaS, Summary of IaaS Providers, Platform as a Service, Characteristics of PaaS, Suitability of PaaS, Pros and Cons of PaaS, Summary of PaaS Providers, Software as a Service, Characteristics of SaaS, Suitability of SaaS, Pros and Cons of SaaS, Summary of SaaS Providers, Other Cloud Service Models, Scaling Basics: What is scaling, Basic capacity planning **CO's-CO4**

Self-Learning Topics: Cloud SDK usage and simple application deployment concepts

UNIT V: **12 Hours**

Cloud Service Providers: EMC, EMC IT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue, and service, Microsoft, Microsoft Assessment and Planning Toolkit. Cloud Application Platform (Aneka Basics): Platform Overview, Introduction to Aneka.

Self-Learning Topics: Server fewer computing basics (Function as a Service concept)

CO's-CO5

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

TEXT BOOK:

1. Essentials of cloud Computing: K. Chandrasekharan, CRC press, 2014

REFERENCE BOOKS:

1. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, rp 2011.

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Cloud Computing.
2. List the key characteristics of cloud computing.
3. What is virtualization?
4. Name the types of virtualizations.
5. Expand IaaS, PaaS, and SaaS.
6. What is a Public Cloud?
7. Define Scalability in cloud computing.
8. What is Disaster Recovery?
9. What is a Service Level Agreement (SLA)?
10. What is Amazon Web Services?
11. What is Microsoft Azure?

L2: Understand

1. Explain the concept of cloud computing and its key characteristics.
2. Explain how virtualization enables cloud computing.
3. Differentiate between full virtualization and para-virtualization.
4. Explain the architecture of cloud computing with a neat diagram.
5. Differentiate between IaaS, PaaS, and SaaS with examples.
6. Compare public, private, and hybrid cloud models.
7. Explain scalability and fault tolerance in cloud computing.
8. Describe the security challenges in cloud computing and their solutions.
9. Explain disaster recovery in cloud computing with suitable examples.
10. Explain the working and architecture of Aneka platform
11. Compare major cloud platforms like Amazon Web Services, Google App Engine Microsoft Azure.

L3: Apply

1. Apply virtualization concepts to design a small cloud infrastructure for a college lab.
2. Given a startup scenario, choose between IaaS, PaaS, and SaaS and justify your choice.
3. Design cloud architecture for an e-commerce application using appropriate service models.
4. Apply scalability techniques to handle sudden traffic spikes in a web application.
5. How would you implement a disaster recovery plan for a banking system hosted on cloud?
6. Select a suitable cloud deployment model (public/private/hybrid) for a hospital system and justify.
7. Apply security mechanisms to protect sensitive user data in a cloud-based application.
8. Demonstrate how you would deploy an application using Aneka platform.
9. Choose between Amazon Web Services and Microsoft Azure for a real-world project and justify your decision.
10. Apply capacity planning techniques for scaling a cloud-based video streaming service.
11. Design a secure cloud database system with proper access control and backup strategies.

L4: Analysing

1. Analyze how virtualization contributes to resource optimization and cost efficiency in cloud computing.
2. Compare and analyze IaaS, PaaS, and SaaS in terms of control, flexibility, and use cases.
3. Analyze the differences between public, private, and hybrid clouds based on security, cost, and scalability.
4. Examine the cloud computing reference model and analyze the role of each layer.
5. Analyze the impact of scalability and fault tolerance on system performance in cloud environments.
6. Evaluate the security challenges in cloud computing and analyze their potential risks.
7. Analyze different disaster recovery strategies and identify the most suitable approach for critical systems.
8. Analyze the architecture of Aneka and explain how its components interact.
9. Compare and analyze major cloud platforms such as Amazon Web Services, Google App Engine, and Microsoft Azure in terms of services and scalability.
10. Analyze the trade-offs between performance and security in cloud-based applications.
11. Break down a real-world cloud application (e.g., online streaming system) and analyze **its** architecture and components.

L5: Evaluating

1. Evaluate the effectiveness of cloud computing compared to traditional computing models.
2. Assess the role of virtualization in improving cloud performance and resource utilization.
3. Evaluate IaaS, PaaS, and SaaS models and justify which is most suitable for different business scenarios.
4. Critically evaluate public, private, and hybrid cloud deployment models for enterprise use.
5. Assess the economic benefits and hidden costs of adopting cloud computing.
6. Evaluate the security and privacy concerns in cloud computing and suggest improvements.
7. Critically evaluate disaster recovery strategies in cloud environments for mission-critical applications.
8. Assess the effectiveness of Aneka as a cloud application development platform.
9. Evaluate and justify the selection of a cloud platform among Amazon Web Services, Google App Engine, and Microsoft Azure for a given real-world scenario.
10. Evaluate the challenges of cloud interoperability and propose suitable solutions.
11. Assess the trade-offs between cost, performance, and security in cloud-based systems.

L6: Create

1. Design a complete cloud computing architecture for a smart university system including services, storage, and security.
2. Propose a cloud-based e-commerce system with proper use of IaaS, PaaS, and SaaS models.
3. Design a disaster recovery framework for a banking system using cloud infrastructure.
4. Create a scalable cloud architecture for a video streaming platform like Netflix or YouTube.
5. Design a secure cloud-based healthcare system ensuring privacy, data security, and compliance.
6. Propose a hybrid cloud solution for a multinational company and justify its architecture.
7. Design a cloud deployment strategy using Aneka for distributed scientific computing applications.
8. Create an architecture using Amazon Web Services services for a real-time analytics system.

9. Design a cloud-based learning management system (LMS) with scalability and fault tolerance features.
10. Propose an optimized cloud solution comparing and integrating services from Google App Engine and Microsoft Azure.
11. Design a cost-efficient cloud infrastructure model that balances performance, scalability, and security for a startup.

**Chairperson
Board of Studies (CSE)**

Course Objectives:

1. To provide student with an academic environment aware of various AI Algorithms.
2. To train Students with python programming as to comprehend, analyze, design and create AI platforms and solutions for the real-life problems.
3. Learn usage of Libraries for Machine Learning in Python
4. Describe appropriate supervised/Unsupervised learning algorithms for a given problem

Course Outcomes:

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

Course Code	Course Outcomes	Mapping with POs and PSOs													
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	Do K
R24CSPC 23.1	Implement and analyze classical AI search algorithms and problem-solving techniques	3	3	2	2	2	-	-	-	1	-	2	3	2	L1, L2
R24CSPC 23.2	Apply data preprocessing, visualization, and statistical learning methods using Python libraries.	3	2	2	2	3	-	-	-	1	-	2	2	3	L2, L3
R24CSPC 23.3	Design and implement machine learning models and neural networks for classification, regression, and clustering tasks.	3	3	3	3	3	-	-	-	2	-	3	3	3	L3

Developing the following programs using Python**Week 1:**

CO's-CO1

1. a) Write a program to implement Breadth First Search.
- b) Write a program to implement Depth First Search.

Week 2:

CO's-CO1

2. Write a program to implement Hill Climbing Algorithm.

Week 3:

CO's-CO1

3. Write a program to implement A* Algorithm.

Week 4:

CO's-CO1

4. Write a program to implement Tic-Tac-Toe game.

Week 5:

CO's:CO1

5. Write a program to implement Water Jug Problem.

Week 6:

CO's-CO2

6. a) Write a python program to import and export data using Panda's library functions.
- b) Demonstrate various data pre-processing techniques for a given dataset.

Week 7:

CO's-CO2

7. a) Implement Dimensionality reduction using Principal Component Analysis (PCA) method.
b) Write a Python program to demonstrate various Data Visualization Techniques.

Week 8: CO's-CO3

8. a) Implement Simple and Multiple Linear Regression Models.
b) Develop Logistic Regression Model for a given dataset.

Week 9: CO's-CO3

9. Develop Decision Tree Classification model for a given dataset and use it to classify a new sample.

Week 10: CO's-CO3

10. a) Implement Naïve Bayes Classification in Python
b) Build KNN Classification model for a given dataset.

Week 11: CO's-CO3

11. Build Artificial Neural Network model with back propagation on a given dataset.

Week 12: CO's-CO3

12. Write a python program to implement K-Means clustering Algorithm.

Board of Studies : Computer Science and Engineering

Approved in BOS No: 03

Approved in ACM No: 03

Reference Books:

1. Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach", Pearson Education
2. Tom M. Mitchell, "Machine Learning, McGraw-Hill
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor Flow", O'Reilly
4. Jake VanderPlas, "Python Data Science Handbook", O'Reilly

Online Learning Resources:

1. <https://nptel.ac.in>
2. <https://www.coursera.org>

**Chairperson
Board of Studies(CSE)**

R24CSPC24

CRYPTOGRAPHY NETWORK SECURITY LAB 0 0 3 1.5

(Computer Science and Engineering)

Course Objectives:

1. To understand the basic principles of cryptography and network security.
2. To study and implement classical and modern encryption algorithms.
3. To develop programming skills in C and Java for cryptographic applications.
4. To apply mathematical concepts like modular arithmetic in security algorithms.
5. Understand various cryptographic algorithms.

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs													
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	DoK
R24CSPC 24.1	Apply basic programming and bitwise operations to manipulate strings using C language.	3	2	2	-	3	1	-	1	1	-	1	3	2	L1, L2
R24CSPC 24.2	Implement and demonstrate classical and symmetric encryption algorithms such as Caesar Cipher, Substitution Cipher Hill Cipher, DES, Blowfish, and Rijndael (AES).	3	2	3	2	3	1	1	1	1	-	2	3	3	L2, L3
R24CSPC 24.3	Analyze and implement modern cryptographic techniques including public key algorithms (RSA, Diffie-Hellman) and hashing techniques (SHA-1) for secure communication and data integrity.	3	3	2	3	3	2	3	1	1	1	2	3	3	L3

Developing the following programs:

WEEK 1

1. Write a C program that contains a string (char pointer) with a value 'Hello World'. The program should XOR each character in this string with 0 and displays the result.

CO's-CO1

WEEK 2

2. Write a C program that contains a string (char pointer) with a value 'Hello World'. The program should AND or and XOR each character in this string with 127 and display the result

CO's-CO1

WEEK 3

3. Write a Java program to perform encryption and decryption using the following algorithms:

- a) Ceaser Cipher
 - b) Substitution Cipher
 - c) Hill Cipher .
- CO's-CO1**

WEEK 4

- 4. Write a Java program to implement the DES algorithm logic
- CO's-CO2**

WEEK 5

- 5. Write a C/JAVA program to implement the BlowFish algorithm logic
- CO's-CO2**

WEEK 6

- 6. Write a C/JAVA program to implement the Rijndael algorithm logic.
- CO's-CO2**

WEEK 7

- 7. Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool.
- CO's-CO2**

WEEK 8

- 8. Write a Java program to implement RSA Algorithm
- CO's:CO2**

WEEK 9

- 9. Implement GCD, Modular Inverse, and Euler's Totient Function
- CO's-CO3**

WEEK 10

- 10. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
- CO's-CO3**

WEEK 11

- 11. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
- CO's-CO3**

WEEK 12

- 12. Calculate the message digest of a text using the MD5 algorithm in JAVA.
- CO's-CO3**

WEEK 13

- 13. Write a Java program to implement Digital Signature using RSA Algorithm for signing and verification of a message.
- CO's-CO3**

WEEK 14

- 14. Write a Java program to implement Secure Socket Communication using SSL/TLS.
- CO's-CO3**

Board of Studies : Computer Science and Engineering

Approved in BOS No : 03

Approved in ACM No: 03

Text books:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition.
2. Cryptography and Network Security: Atul Kahate, McGraw Hill, 3rd Edition.

Reference books:

1. Cryptography and Network Security: C K Shyamala, N Harini, Dr T R Padmanabhan, Wiley India, 1st Edition.
2. Cryptography and Network Security: Forouzan Mukhopadhyay, McGraw Hill, 3rd Edition.
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH.
5. Introduction to Network Security: Neal Krawetz, CENGAGE Learning.
6. Network Security and Cryptography: Bernard Menezes, CENGAGE Learning

Online Learning Resources:

1. <https://cacr.uwaterloo.ca/hac/>
2. <https://williamstallings.com/Cryptography/>

**Chairperson
Board of Studies (CSE)**

R24CSSC04

PROMPT ENGINEERING

0 1 2 1.5

(Skill Oriented Course)

Course Objective:

1. Equip learners with the skills to design, optimize, and evaluate prompts for diverse tasks using LLMs in real-world applications.

Course Outcomes:

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSSC 04.1	Understand basic concepts of prompt engineering and create simple prompts for different tasks like text generation, classification, and summarization.	2	2	1	-	2	-	-	-	1	-	1	2	2	L1, L2, L3
R24CSSC 04.2	Use advanced prompting techniques and improves prompts by analyzing and evaluating the responses of language models.	2	3	2	2	3	1	-	1	2	-	2	2	3	L3, L4, L5
R24CSSC 04.3	Build simple applications using APIs and prompts and apply their knowledge to solve real-world problems.	2	2	3	1	3	1	-	1	2	-	2	3	3	L3, L4

SYLLABUS**Week-1:****COs-CO1**

Introduction to Prompt Engineering: Definition of prompt & Prompt Engineering, Types of language models (ChatGPT, Claude, Gemini, etc.), Prompting paradigms: zero-shot, few-shot, chain-of-thought?

Activity:

1. Exploring ChatGPT or Open AI Playground
2. Basic prompt crafting and analysis

Week-2:**COs-CO1****Foundations of Prompt Crafting**

Role of clarity, context, and constraints, Prompt templates and best practices, Role-playing and persona-based prompting of training data in LLM performance

Activity:

1. Rewrite/improve prompts for clarity and output quality
2. Use templates for different tasks (e.g., summarize, explain, translate)

Week-3:**COs-CO1****Task-Specific Prompting**

Categories-Text generation (creative writing, marketing), Classification and extraction (e.g., sentiment analysis, data extraction), Question answering and summarization

Activity:

1. Design prompts for each category
2. Evaluate and iterate based on outputs

Week-4 and 5:

COs-CO2

Advanced Prompting Techniques

Chain-of-thought reasoning, Step-by-step breakdowns, Self-consistency and verification, System prompts and role instructions

Activity:

1. Use multi-step prompts for logical reasoning or math
2. Build a guided assistant with structured response flows

Week-6 and 7:

COs-CO2

Prompt Evaluation & Iteration

Evaluation metrics (fluency, relevance, coherence), A/B testing and prompt tuning, Limitations of LLMs and prompt robustness

Activity:

1. Compare prompt variants
2. Analyze hallucinations and mitigate errors

Week-8 and 9:

COs-CO3

APIs and Automation

Using the Open AI API (or other LLM APIs), Prompt injection defines basics, Automating tasks with prompts (scripts, tools)

Activity:

1. Build a simple prompt-based app (e.g., email assistant or quiz generator)
2. Use prompt chaining via API

Week-10:

COs-CO3

Capstone Project

Design a prompt-based solution for a real-world problem:

1. Examples: résumé optimizer, content generator, legal contract explainer

Board of Studies : Computer Science and Engineering

Approved in BoS No :03

Approved in ACM No :03

Text Books:

1. John Berryman & Albert Ziegler, Prompt Engineering for LLMs the Art and Science of Building Large Language Model–Based Applications, OREILLY publication, 2024.

Reference Books:

1. James Phoenix, Mike Taylor, Prompt Engineering for Generative AI, O'Reilly Media, Inc., 2024.
2. Padmaraj Nidagundi, Prompt Engineering: A Handbook for Prompt Engineering, NLP Engineers, Software developers, AI Tech leaders, Startup business owners and Other IT Professionals, Kindle Edition, 2022.

Web References:

1. [Prompt Engineering Guide | Prompt Engineering Guide](#)
2. [Prompt Engineering for AI Guide | Google Cloud](#)
3. [Prompt Engineering Guide: The Ultimate Guide to Generative AI](#)
4. [What is Prompt Engineering - Meaning, Working, Techniques - GeeksforGeeks](#)

**Chairperson
Board of Studies (CSE)**

R24HS09

High Level Reasoning & Employability Skills

0 0 2 1

(Common to All)

Course Objectives:

1. **Develop professional skills:** Students will be able to apply professional etiquette and effective problem-solving processes to make ethical decisions in workplace settings.
2. **Enhance interview skills:** Students will be able to Develop professional career artifacts, including resumes and cover letters, while demonstrating effective interview and networking skills.
3. **Develop logical thinking skills:** Students will be able to Solve advanced logical problems involving Syllogisms, Venn Diagrams, Cubes & Dices, and Coded Inequalities.
4. **Improve problem-solving skills:** Students will be able to Analyze complex seating arrangements, data sufficiency, and mathematical operations to derive logical conclusions.
5. **Enhance analytical skills:** Students will be able to Evaluate analytical reasoning puzzles and critical reasoning statements, including assumptions, arguments, and cause-effect relationships.

Course Outcomes:

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs													Dok
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24HS09.1	Professional Etiquette & Decision Making	-	-	-	-	-	-	1	-	2	-	-	-	-	L3
R24HS09.2	Career Prep & Interview Skills	-	-	-	-	-	-	-	-	3	-	2	-	-	L3
R24HS09.3	Logical Thinking (Syllogism, Venn, Cubes)	3	-	-	-	-	-	-	-	-	-	-	-	-	L3
R24HS09.4	Problem Solving (Arrangements & Data Sufficiency)	3	-	-	-	-	-	-	-	-	-	1	-	-	L4
R24HS09.5	Analytical Reasoning & Critical Thinking	2	-	-	-	-	-	-	-	-	-	1	-	-	L5

SYLLABUS**UNIT-I: Professional Etiquette & Ethics****13 Hours**

Problem-Solving and Decision-Making: The problem-solving process (identifying, analyzing, generating solutions), Critical and creative thinking, Making effective and ethical decisions. Professional Etiquette and Manners: Workplace etiquette (cubicle, meeting, and break room behavior), Telephone and virtual meeting etiquette, Interacting with clients and superiors.

CO's-CO1**Self Learning Topic:** Corporate Etiquette Simulation

UNIT-II: Interview Skills & Career Planning **12 Hours**

Career Preparation: Developing a Resume/CV and cover letter, interviewing skills (pre-interview preparation, common questions, post-interview follow-up), Networking and building professional relationships. **CO's-CO2**

Self-Learning Topic: Mock Interviews and GD Practice

UNIT-III: Logical Thinking Skills **13 Hours**

Syllogism: Explain Advanced Concepts on Universal Statements and Particular Statements. Venn Diagrams: Explain Advanced Concepts on group-based problems and datatype and diagram-based problems. Cubes & Dices and Coded Inequalities. **CO's-CO3**

Self-Learning Topic: Sequence & Alphabet Test

UNIT-IV: Problem Solving Skills **15 Hours**

Seating Arrangements: Explain advanced problems on Linear, Circular and Square based problems. Data Sufficiency and Mathematical Operations. **CO's-CO4**

Self-Learning Topic: Non-Verbal Reasoning

UNIT-V: Analytical Thinking Skills **12 Hours**

Analytical Reasoning: Explain advanced problem solving and puzzle-based questions, Critical Reasoning: Explain advanced problems on Statements-Conclusions, Statements-Assumptions, Statements-Arguments, Statements-Course of Action and Cause-effect. **CO's-CO5**

Self-Learning Topic: Eligibility Criteria

Board of Studies : MBA

Approved in BoS No : 03

Approved in ACM No: 03

Text Books:

1. Quantitative Aptitude for competitive examinations by R.S.Agrawal - S.Chand publications.
2. Business Communication by C.S.G Krishnamacharyulu & Lalitha Ramakrishna – Himalaya publishing house
3. Barun K. Mitra, Personality Development and Soft Skills, Oxford University Press.

Reference Books:

1. Numerical Aptitude and Reasoning Ability tests in Competitive exams by Prof.A.Balasubramanian.
2. Numerical Ability and Mathematical Aptitude by Dr.A.B.Rao
3. Communication skills and soft skills by E.Suresh Kumar , p.sreehari & j.savithri

Web References:

1. <https://faceprep.in/logical> reasoning
2. www.sawaal.com
3. <http://nptel.ac.in/courses>
4. www.prepinsta.com

Model Questions from High Level Reasoning & Employability Skills

L1: Remember

1. Compare and contrast the leadership styles of a democratic leader and an authoritarian leader. Which style is more effective in a diverse team?
2. Analyze the following conversation and identify the communication barriers present:

Manager: 'Just do whatever you think is best.' Employee: 'I'm not sure what you want me to do exactly.'

3. A candidate performed well technically but failed in the HR interview. Analyze the possible soft skills gaps and suggest improvements.
4. How do group dynamics and individual behavior influence the decision-making process of a team? Provide examples.
5. Evaluate the role of empathy in conflict resolution within a workplace team that includes members from diverse cultural backgrounds.
6. Identify the logical errors in the following group discussion argument: 'Young employees should not be given leadership roles because they lack experience in all situations.'

L2: Understand

1. Critically evaluate the importance of soft skills versus technical skills in long-term career success. Support your argument with real-world examples.
2. A job applicant's resume lists 'excellent communication skills' but struggles during the mock interview. Evaluate what specific areas need improvement and design a remediation plan.
3. Assess the effectiveness of the following interview response to 'Tell me about yourself:' and suggest improvements: 'I am a hardworking student. I like coding and I have done many projects. I want a good job.'
4. Evaluate the impact of poor time management on an employee's professional growth and team performance. Propose strategies to overcome this.
5. A group discussion panel observed that one participant dominated the conversation while others were silent. Evaluate the group dynamics and suggest how the facilitator should manage the situation.
6. Compare the effectiveness of verbal feedback versus written feedback in a professional training context. Which approach is more likely to improve employee performance and why?
7. Design a comprehensive self-improvement plan for a student who wants to develop strong interview skills, professional etiquette, and emotional intelligence within three months.

L3: Apply

1. Statements: Some dogs are bags. No bag is lion. All rooms are lions.
2. Conclusions: I. Some rooms are bags. II. Some dogs are lions. III. Some rooms are dogs.
3. Statements: All rings are birds. Some birds are cages. All cages are kites.
4. Conclusions: I. All kites are cages. II. Some kites are rings. III. Some birds are kites.
5. Statements: Some buses are houses. All houses are taxis. All rickshaws are taxis.
6. Conclusions: I. Some rickshaws are houses. II. Some taxis are houses. III. Some taxis are buses.
7. Statements: All ducks are pigeons. All pigeons are crows.
8. Conclusions: I. Some crows are ducks. II. Some crows are pigeons.
9. A survey was conducted on 70 students regarding the subjects they like. According to the survey, 31 students like Math, 36 like English, and 27 like History. Also 19 like Math and English, 11 like Math and History, and 5 like History and English. 3 students like all the three subjects. How many students do not like any of the three subjects?
10. In a class of 80 students, 37 study French, 62 study German, and everyone in the class studies either French or German. How many of the students study French but not German?
11. Given signs signify something and on that basis, assume the given statements to be true and

find which of the two conclusions I and II is/are definitely true

12. $A+B$ means A is equal to B, $A-B$ means A is not equal to B, $A=B$ means A is greater than B

13. $A*B$ means A is greater than equal to B, A/B means A is less than B

14. Statements: S/T , $T+U$, $U*V$

15. Conclusions: I. S/U II. $T*V$

16. A cube is colored red on all faces. It is cut into 64 smaller cubes of equal size. Now, answer the following questions based on these statements. How many cubes have no face colored?

L4: Analyzing

1. Six friends - P, Q, R, S, T, and U (not necessarily in the same order) are sitting around a circular table, all facing the center. P and Q are not facing each other. Q is sitting second to the left of P and third to the right of R. T is sitting second to the left of S. Who is sitting fourth to the left of S?
2. Five cars parked in a row facing Eastward. E is parked to the left of A, B and C. B, C and A are parked to the left of D, C is parked between A and B. if B is parked fourth from the left, how far is A parked from the right?
3. Six friends Amit, Bhuwan, Chetan, Dilip, Lalit and Fahim are sitting in a row facing west direction. Chetan is between Amit and Lalit. Bhuwan is to the immediate left of Lalit and to the immediate right of Dilip. Fahim is not at the left end. Which pair is sitting beside Dilip?
4. The given question is followed by two statements I and II. Decide if the statements contain sufficient information to answer the question. Problem Question: Among the four students of grade 5 - Richie, Martha, Kelly, and Gretchen, who is the tallest?
5. Statements: I) Martha is shorter than Richie but taller than Gretchen.
6. II) Kelly is taller than Martha.
7. Problem Question: M, P, A and K are four boxes having different weights. Which among them is the heaviest box? Statements: I. P is the heavier than M and K but lighter than A.
8. II. M is lighter than P and A but heavier than K.

L5: Evaluating

1. Statement: Population increase coupled with depleting resources is going to be the scenario of many developing countries in days to come.
2. Conclusions:
 - a. The population of developing countries will not continue to increase in future.
 - b. It will be very difficult for the governments of developing countries to provide decent quality of life.
3. Statement: Facebook is a company committed to giving the people power to build communities and closer together.
4. Assumptions: I. The headquarter of Facebook is in California, United states.
5. II. Facebook is an American social media conglomerate corporation.
6. Statement: In the recent census report, it is stated that the female to male ratio is alarmingly low.
7. Courses of Action: I. The government should conduct another census to verify the results
8. II. The government should immediately issue orders to all the departments to encourage people to improve the ratio.
9. Direction: Read the following information carefully and answer the questions given below.
10. Five friends P, Q, R, S and T stay in a hostel but indifferent wings i.e. wing A, wing B, wing C, wing D and wing E. All of them watch different TV channels Following is some

more information.

11. 'P' does not watch Star Plus or Discovery channel but stays in wing A.
12. 'Q' does not watch CNN.
13. 'S' stays in wing C but he is not interested in watching Zee or Star plus channel.
14. 'R' watches ETV and does not stay in wing D.
15. The person who stays in wing E watches the CNN channel.
16. The friend who stays in wing C watches which channel?

Chairperson
Board of Studies (MBA)

R24ES13

Design Thinking & Innovation
(Common to All Branches)

0 1 2 2

Course Objectives:

1. The objective of this course is to familiarize students with design thinking process as a tool for breakthrough innovation.
2. It aims to equip students with design thinking skills and ignite the minds to create innovative ideas, develop solutions for real-time problems.

Course Code	Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO9	PO10	PO11	PSO1	PSO2	DOK
R24ES13.1	Define the concepts related to design thinking. Explain the fundamentals of Design Thinking and innovation	3	3	2	2	2	-	1	2	2	2	L1, L2
R24ES13.2	Apply the design thinking techniques for solving problems in various sectors.	3	3	3	2	3	-	1	2	3	3	L3
R24ES13.3	Analyse to work in a multidisciplinary environment	3	3	3	2	3	-	-	2	3	3	L4
R24ES13.4	Evaluate the value of creativity	3	2	3	2	3	1	1	3	3	3	L5
R24ES13.5	Formulate specific problem statements of real time issues	3	2	3	2	3	1	1	3	3	3	L3, L6

SYLLABUS**UNIT-I****10 Hours**

Introduction to Design Thinking: Introduction to elements and principles of Design, basics of design-dot, line, shape, form as fundamental design components. Principles of design. Introduction to design thinking, history of Design Thinking, New materials in Industry.

CO's: CO1**UNIT-II****10 Hours**

Design Thinking Process: Design thinking process (empathize, analyze, idea & prototype), implementing the process in driving inventions, design thinking in social innovations. Tools of design thinking - person, costumer, journey map, brainstorming, product development

Activity: Every student presents their idea in three minutes, every student can present design process in the form of flow diagram or flow chart etc. Every student should explain about product development.

CO's: CO2**UNIT-III****10 Hours**

Innovation: Art of innovation, Difference between innovation and creativity, role of creativity and

innovation in organizations- Creativity to Innovation- Teams for innovation- Measuring the impact and value of creativity.

Activity: Debate on innovation and creativity, Flow and planning from idea to innovation, Debate on value-based innovation.

CO's: CO3

UNIT IV

10 Hours

Product Design: Problem formation, introduction to product design, Product strategies, Product value, Product planning, product specifications- Innovation towards product design- Case studies.

Activity: Importance of modeling, how to set specifications, Explaining their own product design.

CO's: CO4

UNIT V

10 Hours

Design Thinking in Business Processes: Design Thinking applied in Business & Strategic Innovation, Design Thinking principles that redefine business – Business challenges: Growth, Predictability, Change, Maintaining Relevance, Extreme competition, Standardization. Design thinking to meet corporate needs- Design thinking for Startups- Defining and testing Business Models and Business Cases- Developing & testing prototypes.

Activity: How to market our own product, about maintenance, Reliability and plan for startup.

CO's: CO5

Board of Studies: Mechanical Engineering

Approved in BOS No: 02,13th May, 2025

Approved in ACM No: 02

Text Books:

1. Tim Brown, Change by design, Harper Bollins (2009)
2. Idris Mootee, Design Thinking for Strategic Innovation, 2013, John Wiley & Sons..

Reference Books:

1. David Lee, Design Thinking in the Classroom, Ulysses press
2. Shruti N Shetty, Design the Future, Norton Press
3. William Lidwell, Universal Principles of Design- Kritinaholden, Jill Butter.
4. Chesbrough. H, The Era of Open Innovation – 2013

Online Learning Resources:

- <https://nptel.ac.in/courses/110/106/110106124/>
- <https://nptel.ac.in/courses/109/104/109104109/>
- https://swayam.gov.in/nd1_noc19_mg60/preview

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1 (%)	Internal Assessment #2 (%)
-----------------	----------------------------	----------------------------

L1	10	--
L2	30	--
L3	30	20
L4	30	30
L5	--	30
L6	--	20
Total (%)	100	100

**Chairperson
Board of Studies (ME)**

R24AC01

Technical Paper Writing and IPR
(Audit Course-1)

2 0 0 0

Course Objectives:

1. To develop the ability to write clear, structured, and professional technical documents.
2. To understand grammar, style, and formatting required for technical writing.
3. To learn drafting, reviewing, and presenting technical papers.
4. To gain proficiency in using word processing tools for documentation.
5. To understand Intellectual Property Rights (IPR) and patenting process.

Course Outcomes

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24AC01.1	Demonstrate ability to write clear and structured technical documents	-	-	-	-	2	-	-	-	-	3	2	2	-	L1, L2
R24AC01.2	Apply drafting and editing techniques in report writing	-	2	-	-	3	-	-	-	-	3	2	2	-	L2, L3
R24AC01.3	Demonstrate proofreading and summarization skills	-	2	-	-	2	-	-	-	-	3	-	2	-	L3, L4
R24AC01.4	Use advanced MS Word tools for technical documentation.	-	-	-	-	3	-	-	-	-	2	-	2	2	L4, L5
R24AC01.5	Understand IPR concepts and patenting process	2	2	-	-	-	2	2	3	-	2	-	-	-	L5, L6

SYLLABUS**UNIT I:****10 Hours**

Introduction to Technical Writing: Introduction to technical writing, technical sentences formation, use of transitions, use of tenses in technical writing.

Planning and Structuring: Planning reports, identifying audience, voice, formatting and structuring reports, sections of a technical report, minutes of meeting writing. **CO's-CO1**

Self-Learning Topics: Case studies of technical reports.

UNIT II:**6 Hours**

Drafting and Editing Techniques: Drafting reports: Use of drafts, illustrations and graphics,

Final Editing: Grammar, spelling, readability, writing in plain English, jargon usage, punctuation, paragraph writing, avoiding ambiguity. **CO's-CO2**

Self-Learning Topics: Writing technical blogs.

UNIT III:**8 Hours**

Proofreading and Presentation Skills: Proofreading techniques, summarization methods, practice exercises.

Presentation Skills: Printed presentation, verbal presentation, introduction to proposals.

Self-Learning Topics: Poster and paper presentation techniques.

CO's–CO3

UNIT IV:

8 Hours

Word Processing Tools for Documentation: Using MS Word: Table of contents, indexing, outlines, comments, track changes. Footnotes, endnotes, citations, bibliography, Document comparison, document protection, macros, formatting tools.

COs–CO4

Self-Learning Topics: Latex basics.

UNIT V:

9 Hours

Intellectual Property Rights (IPR):

Nature of Intellectual Property: Patents, copyrights, trademarks, designs.

Process of Patenting: Research, innovation, patent filing, development.

International Scenario: Global IPR policies and cooperation.

CO's–CO5

Self-Learning Topics: Case studies on patent disputes

Board of Studies : Computer Science and Engineering

Approved in BoS No : 03

Approved in ACM No: 03

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Research Paper Writing using IEEE Format CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Tools for plagiarism detection (Turnitin, Grammarly) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
3. Patent filing in startups and innovation ecosystems CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
4. Role of IPR in software industry, CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2, PSO3, PSO4

Text Books:

1. Technical Communication – Meenakshi Raman & Sangeeta Sharma
2. English for Writing Research Papers – Adrian Wallwork

Reference Books:

1. Day R – *How to Write and Publish a Scientific Paper*
2. MLA Handbook for Writers

Web References:

1. SWAYAM Courses on Technical Writing
2. NPTEL Technical Communication
3. IEEE Author Guidelines

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35

L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is technical writing?
2. Define Intellectual Property (IP).
3. What is a technical report?
4. List the sections of a technical report.
5. What is proofreading?
6. Define plagiarism.
7. What is a patent?
8. What is a trademark?
9. What is copyright?
10. What is meant by citation?

L2: Understand

1. Explain the structure of a technical report.
2. Describe the importance of technical writing in engineering.
3. Explain the role of grammar in technical writing.
4. Differentiate between patent and copyright.
5. Explain the concept of plagiarism and how to avoid it.
6. Describe the process of proofreading.
7. Explain the importance of formatting in technical documents.
8. Describe different types of Intellectual Property Rights.
9. Explain the importance of citations and references.
10. Summarize the process of writing a research paper.

L3: Apply

1. Prepare a short technical report on a given topic.
2. Apply proper formatting to a technical document.
3. Demonstrate proofreading on a sample paragraph.
4. Write a summary of a given technical article.
5. Apply citation styles (IEEE/APA) to a document.
6. Draft minutes of a meeting.
7. Prepare a simple research abstract.
8. Use MS Word tools to create a table of contents.
9. Apply grammar correction to a given passage.
10. Create a bibliography for a given set of references.

L4: Analysing

1. Analyze the structure of a research paper.
2. Compare technical writing with general writing.
3. Analyze errors in a poorly written technical document.
4. Compare different citation styles (IEEE vs APA).
5. Analyze the importance of clarity and conciseness.

6. Examine the role of visuals in technical reports.
7. Analyze the patenting process step by step.
8. Compare different types of Intellectual Property Rights.
9. Analyze ethical issues related to plagiarism.
10. Examine formatting issues in a sample document.

L5: Evaluating

1. Evaluate the importance of technical writing in professional communication.
2. Assess the effectiveness of a given technical report.
3. Evaluate the importance of IPR in the software industry.
4. Judge the quality of a research paper based on structure.
5. Evaluate different plagiarism detection tools.
6. Assess the role of patents in innovation.
7. Evaluate the effectiveness of proofreading techniques.
8. Compare advantages of different citation styles.
9. Evaluate the impact of IPR laws on startups.
10. Justify the need for clear documentation in engineering.

L6: Create

1. Draft a complete technical report on a given topic.
2. Create a research paper using proper structure and formatting.
3. Design a presentation for a technical paper.
4. Develop a proposal for a technical project.
5. Create a plagiarism-free document with proper citations.
6. Design a template for technical report writing.
7. Develop a document using advanced MS Word features.
8. Create a case study on IPR issues in software development.
9. Draft a patent application outline for an innovation.
10. Design a complete documentation system for a project.

**Chairperson
Board of Studies (CSE)**