गोंय विद्यापीठ
ताळगांव पठार,
गोंय - ४०३ २०६
फोन : + ९१ - ८६६९६०९०४८

Goa University
Taleigao Plateau,Goa - 403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website : www.unigoa.ac.in

(Accredited by NAAC with Grade A+)

GU/Acad –PG/BoS -NEP/2026-27/224 Date: 20.07.2026

CIRCULAR

Ref. No.: GU/Acad –PG/BoS -NEP/2024/508 dated 12.09.2024,
GU/Acad –PG/BoS -NEP/2026-27/223 dated 20.07.2026

In continuation to the above referred Circulars, the syllabus for Semester V & VI of the **Bachelor of Engineering in Computer Engineering** Programme as approved by the Academic Council in its meeting held on 20th February 2026 is attached.

The Dean, Faculty of Engineering and Principals of affiliated Colleges offering the **Bachelor of Engineering in Computer Engineering** Programme are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

ASHWIN V LAWANDE
Digitally signed by ASHWIN V LAWANDE
Date: 2026.07.20 11:59:23 +05'30'

(Ashwin V. Lawande)
Deputy Registrar – Academic

To,

1. The Dean, Faculty of Engineering, Goa University.
2. The Principals of Affiliated Engineering Colleges.

Copy to:

1. Chairperson, BoS Computer Engineering, Goa University.
2. Controller of Examinations, Goa University.
3. Assistant Registrar Examinations (Technical and Allied), Goa University
4. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

COMPUTER ENGINEERING (AY 2024-2025)

SEMESTER – V							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**CMP-300**	Database Management Systems	3	0	0	3
		CMP-301	Database Management Systems Lab	0	0	1	1
2	**Major**	**CMP-302**	Data Communications and Computer Networks	3	0	0	3
		CMP-303	Data Communications and Computer Networks Lab	0	0	1	1
3	**Major**	**CMP-304**	Artificial Intelligence	3	0	0	3
		CMP-305	Artificial Intelligence Lab	0	0	1	1
4	**Major**	**CMP-306**	Design and Analysis of Algorithms	2	0	0	2
		CMP-307	Design and Analysis of Algorithms Lab	0	0	2	2
5	**Prof. Electives**	**CMP-321**	Software Testing and Quality Assurance	3	0	0	3
		CMP-322	Software Testing and Quality Assurance Lab	0	0	1	1
		OR					
		CMP-323	Blockchain Technology	3	0	0	3
		CMP-324	Blockchain Technology Lab	0	0	1	1
		OR					
		CMP-325	Soft Computing	3	0	0	3
		CMP-326	Soft Computing Lab	0	0	1	1
			TOTAL	**14**	**0**	**6**	**20**

Approved by: Academic Council on 20th February 2026

SEMESTER – VI							
Sr. No.	**Course Category**	**Course Code**	**Title of the Course**	**L**	**T**	**P**	**TCr**
1	**Major**	**CMP-308**	Virtualization and Cloud Computing	3	0	0	3
		CMP-309	Virtualization and Cloud Computing Lab	0	0	1	1
2	**Major**	**CMP-310**	Operating Systems	3	0	0	3
		CMP-311	Operating Systems Lab	0	0	1	1
3	**Major**	**CMP-312**	Machine Learning	2	0	0	2
		CMP-313	Machine Learning Lab	0	0	2	2
4	**Major**	**CMP-314**	Image Processing	3	0	0	3
		CMP-315	Image Processing Lab	0	0	1	1
5	**Prof. Electives**	**CMP-327**	Advanced Computer Graphics	3	0	0	3
		CMP-328	Advanced Computer Graphics Lab	0	0	1	1
		OR					
		CMP-329	Software Defined Networks	3	0	0	3
		CMP-330	Software Defined Networks Lab	0	0	1	1
		OR					
		CMP-331	GPU Architecture and Programming	3	0	0	3
		CMP-332	GPU Architecture and Programming Lab	0	0	1	1
			TOTAL	**14**	**0**	**6**	**20**

SEMESTER V

Major Courses

Name of the Programme **: B.E. in Computer Engineering**

Course Code **: CMP-300**

Title of the Course **: Database Management Systems**

Number of Credits **: 3(3L)**

Effective from AY **: 2024-25**

Prerequisites for the course:	Basics of Data Structures	
Course Objectives:	**This course will enable students to:** 1. Understand the Basics of Database Management Systems (DBMS) and data modeling. 2. Apply SQL queries to relational databases and manage and manipulate data within a relational database system, and the ability to design and build a database system. 3. Analyze query processing, transaction processing, including the principles of concurrency control and ensuring data integrity in multi-user environments. 4. Optimize queries and explore NoSQL databases, with a focus on MongoDB, to address the needs of modern applications requiring scalable and flexible data storage solutions.	
Content:		**No. of Hours**
Unit-1	**Introduction**: Characteristics of Database approach, advantages of using the DBMS approach, Three schema architecture, Data Models. **Entity–Relationship Model**: Entity –Relationship Model, Constraints, removing redundant attribute in entity set, Entity-Relationship diagram, reduction to relational schema, Extended ER features. **The Relational Model:** Relational model concepts, Constraints, and relational Database schema. **Relational Algebra:** Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from Set theory Binary Relational Operations: JOIN and DIVISION, Aggregate functions, and Grouping.	**11**
Unit-2	**Basic SQL:** SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT, DELETE and UPDATE statements in SQL.	**11**

	More SQL: Complex Queries, Nested Queries, Aggregate Operators, Views, Specifying Constraints as Assertions and Actions as Triggers. **Relational Database Design:** Informal design guidelines for relational schemas, Functional dependencies, Normal forms: 1NF, 2NF, 3NF, BCNF. Database Design Theory: Inference rules, Equivalence and minimal cover.	
Unit-3	**Transaction processing concepts:** Transaction and system concepts, desirable properties of transaction, characterizing schedules based on recoverability, characterizing schedules based on serializability. **Concurrency Control:** Two phase locking technique for concurrency control, concurrency control based on timestamp ordering, Multiversion concurrency control technique, validation concurrency control technique. **Query Processing:** Measures of Query Cost, Selection operation, Sorting, Join operation (Nested-Loop Join, Block Nested Loop join, Indexed Nested-Loop join, Merge Join), Evaluation of Expression.	**11**
Unit-4	**Query Optimization:** Overview, Using heuristics in query optimization **Getting Started with NoSQL**: About NoSQL, Why NoSQL?, SQL versus NoSQL, ACID versus BASE, CAP Theorem. **Introduction to different types of NoSQL Databases:** Key−Value Pair Databases, Document Databases, Column-Family Databases, Graph Database **MongoDB:** Introduction, CRUD Operations, Aggregation, Sharding & Replication in MongoDB.	**12**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TextBooks** 1. Manu Sharma, MongoDB Complete Guide, BPB Publications, 2021, ISBN: 978938989886 2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education, 2018, ISBN: 9788131716250 3. Seema Acharya, Demystify NoSQL, Wiley, 2019, ISBN:9788126579969 **Reference Books** 1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw Hill Education, 2013, ISBN:	

	9780078022159 2. Dan Sullivan, NOSQL for Mere Mortals, Addison-Wesley Professional, Pearson Education, Inc., 2015, ISBN: 978-0134023212. 3. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2002, ISBN: 9780072465631
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Explain the fundamentals of data models, model database systems using ER diagrams, and construct queries in relational algebra. CO2: Apply SQL queries and normalization techniques to relational databases. CO3: Analyze transaction processing, concurrency control mechanisms, and query processing techniques. CO4: Evaluate the differences between NOSQL databases and relational databases and perform query optimization.

Name of the Programme : **B.E. in Computer Engineering**

Course Code : **CMP-301**

Title of the Course : **Database Management Systems Lab**

Number of Credits : **1P**

Effective from AY : **2024-25**

Prerequisites for the course:	Basic Programming skills	
Course Objectives:	**This course will enable students to:** 1. Understand fundamental database concepts and the underlying concepts of database technology 2. Apply SQL programming to relational database systems 3. Create and query databases using MongoDB 4. Develop database applications using frontend tools and back end DBMS	
	List of Experiments	**No. of Hours**
Content:	1. Study of Data Definition Language(DDL) statements 2. Study of Data Manipulation Language (DML) statements 3. Use of SELECT command with various clauses and Implementation of Integrity Constraints (PRIMARY KEY, FOREIGN KEY, etc.) and Use of SET OPERATORS (UNION,INTERSECT,MINUS) 4. Use of GROUP BY with aggregate functions (AVG,COUNT, MAX, MIN, SUM) 5. Study of various types of JOINS (INNER,LEFT,RIGHT,FULL) 6. Study of various nested queries and subqueries 7. Creation and usage of SQL Views 8. Implementation of Triggers in SQL 9. Study of basic MongoDB operations (Insert, Find, Update, Delete) and Implement a MongoDB database and perform CRUD operations 10. Mini Project–Develop a database application with front- end and back-end integration	**30**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**TextBooks** 1. Manu Sharma, MongoDB Complete Guide, BPB Publications, 2021, ISBN: 978938989886	

	2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education, 2018, ISBN: 9788131716250 **Reference Books** 1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw Hill Education, 2013, ISBN: 9789332901384 2. Shakuntala Gupta Edward, Navin Sabharwal, Practical MongoDB: Architecting, Developing, and Administering MongoDB,1st Edition, Apress, 2015, ISBN: 9781484206485.
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Apply Data Definition Language (DDL) and Data Manipulation Language (DML) commands to create, modify, and manage relational database structures and data. CO2: Analyze SQL queries to retrieve and manipulate data efficiently. CO3: Implement backend storage using MongoDB CO4: Design a database application by integrating front-end and back-end.

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-302**

Title of the Course : **Data Communications and Computer Networks**

Number of Credits : **3(3L)**

Effective from AY : **2024-25**

Pre-requisites for the Course:	1. Basic Knowledge of Computer Fundamentals, operating system and Programming. 2. Understanding of Binary and hexadecimal Number Systems	
Course Objectives:	**This course will enable students to:** 1. Develop ability to understand the fundamentals of data communication and Computer networks and disseminate the knowledge of communication of data by using different conversion mechanics 2. Develop an understanding of networking devices and components 3. Learn various error detection, error correction, and flow control mechanisms 4. Critically analyze and appreciate the working routing protocols and standards	
Contents:		**No of Hours**
Unit-1	**Fundamentals of Data Communication and Computer Network** Introduction to Computer Networks, Network types: LAN, WAN, MAN, Process of data communication and its components: Transmitter, Receiver, Medium, Message, Protocols and layered architecture: OSI-ISO/TCP-IP model, Standards, Standard organizations, Bandwidth, Data Transmission Rate, Baud Rate and Bits per second, Modes of Communication (Simplex, Half duplex, Full Duplex) Analog Data/Signal and Digital Data/Signal, Digital Transmission Techniques, **Digital to Digital Conversion:** Line coding, Block coding, Scrambling, **Analog To Digital Conversion:** PCM, Delta modulation, Analog Transmission techniques, **Digital To Analog Conversion:** ASK, FSK, PSK, **Analog to Analog conversion:** AM,FM,PM	**12**
Unit 2	**Transmission Media And Switching** Transmission on Noisy and Noiseless channels and its maximum transmission capacity,**Communication Media:** Guided Transmission Media-Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable , Unguided Transmission Media-Radio Waves, Microwaves, Infrared, Satellite , Line-of-Sight Transmission, Point-to-Point, Broadcast **Multiplexing:** Frequency-Division	**11**

	Multiplexing ,Time - Division Multiplexing Space Division Multiplexing, Wavelength Division Multiplexing **Switching**: Circuit-switched network, Packet switched network, Message switched networks**Network Computing Model:** Peer To Peer, Client Server **Network Topologies:** Introduction, Definition, Selection criteria, Types of Topology- Star ,Mesh, Tree, Hybrid **Network Connecting Devices:** Switch, Router, Repeater, Bridge, Gateways and Modem	
Unit 3	**Data link layer**: Introduction, Framing, and Error – Detection and Correction – Parity – LRC – VRC , CRC, Hamming code, Flow and Error Control protocols, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols. Medium Access sub layer: ALOHA, CSMA/CD, LAN – Ethernet IEEE 802.3, IEEE 802.5 – IEEE 802.11, Random access, Controlled access, Physical Addressing **Network layer:** Logical Addressing, IPv4 Addressing: Classfull and Classless Addressing ,subnet mask, supernetting, subnetting , IPV6 Addressing scheme, Internetworking, Tunneling, Address mapping, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols Internet Layer protocols -IP,ARP,RARP,ICMP, IGMP	**11**
Unit 4	**Transport Layer:** Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, QoS in Switched Networks, UDP/TCP-Socket programming **Application Layer:** Domain name space, DNS in internet, electronic mail, ,WWW, HTTP, SNMP, FTP,HTTP,SMTP,TELNET,BOOTP,DHCP	**11**
Pedagogy:	The teaching-learning process shall integrate interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-analyzing skills	
Readings	**Text Books:** 1. Andrew S Tanenbaum, Computer Networks, 6th Edition. Pearson Education; Pearson India, 2022, ISBN-13 978-9356063600 2. Behrouz A. Forouzan , Data Communications and Networking, Sixth Edition TMH, 2017. ISBN-9780070 634145 **Reference Books:** 1. James F.Kurose& Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 7th Edition, Pearson Education.2017, ISBN 9780133594140 2. P.C. Gupta, Data communications and Computer Networks, 2nd edition PHI learning, 2013,ISBN -9788120348646 3. William Stallings, Data and Computer Communication, Eight Edition, Pearson Education, 2017. ISBN: 9788131715369	

Course Outcomes	**Upon completion of the course, students will be able to:** CO1: Analyze the functioning of Data Communication and Computer Network. CO2: Select relevant Transmission Media and Switching Techniques as per need. CO3: Configure different TCP/IP protocols/services and conduct their performance analysis. CO4: Design and Implement relevant Network Topology using Networking Devices.

Name of the Programme	**: B.E. Computer Engineering**
Course Code	**: CMP-303**
Title of the Course	**: Data Communications and Computer Networks Lab**
Number of Credits	**: 1P**
Effective from AY	**: 2024-25**

Pre-requisites for the Course:	**C/C++/Java/Python, Data structures**	
Course Objectives:	**This course will enable students to:** 1. Understanding the basic communication principles between the computers 2. Constructing the network in different topological structure 3. Constructing and configuring IP networks 4. Analyze the performance of different routing algorithms	
Contents:	**Experiment List** 1. Implement ASK,PSK,FSK using open source simulation tool 2. Create and Test standard straight/crossover/rollover network cable(Universal Colour Code) using crimping 3. Generate TDM/FDM and demultiplex them using Simulation 4. Write a program to implement the following Error detection and correction mechanism a)CRCb)Hamming code c) Checksum 5. Implement different Framing methods and Analyze their efficiency 6. Configure static IP address in operating system along with appropriate subnet mask for given problem 7. Execute TCP/IP network commands and understand its functionality of each command (ipconfig, ping,tracert,netstat, pathping ,route) 8. Create and implement the following topologies using simulators and physical equipment a)Bus b) Star c)Mesh d) Tree e) Ring 9. Implement the following routing algorithms a. Link State Routing b. Distance Vector Routing c. Path Vector Routing 10. Implement The following flow and error control Mechanisms a. Stop and wait	**No of Hours** **30**

	b. Selective repeat c. Go back N	
Pedagogy:	Instructional and Collaborative Learning, Inquiry Based Learning, Experiential Learning.	
Readings	**Text Books:** 1. James F.Kurose& Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 7th Edition, Pearson Education.2017, ISBN 9780133594140 2. P.C .Gupta, Data communications and Computer Networks, 2nd edition PHI learning, 2013,ISBN -9788120348646 **Web References**: 1. https://cn-iitr.vlabs.ac.in/List%20of%20experiments.html 2. https://www.nitttrkol.ac.in/virlab.php#top 3. https://www.netacad.com/cisco-packet-tracer	
Course Outcomes	**After going through this course, the students will be able to:** CO1: Understand the working of multiplexing of channels, as well as encoding/decoding of data transmission on the channels CO2: Explain simple network commands in the command prompt to diagnose and troubleshoot networking connectivity issues CO3: Demonstrate the working of the networking devices in different networking scenarios CO4: Implement different error detection and routing techniques	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-304**

Title of the Course : **Artificial Intelligence**

Number of Credits : **3(3L)**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Fundamental knowledge of programming , data structures, algorithms, and mathematics.	
Course Objectives:	**This course will enable students to:** 1. Apply classical, heuristic, and optimal search strategies to solve AI problems. 2. Develop structured solutions using planning, decomposition techniques, constraint satisfaction, and game-playing algorithms. 3. Demonstrate usage of propositional logic, first-order logic, structured way of knowledge representation, and expert system concepts. 4. Explain no-code/low-code AI systems, edge computing, and transfer learning concepts.	
Contents:		**No of Hours**
Unit - 1	**Introduction, State Space Search and Heuristic Search** **Artificial Intelligence:** Introduction, State Space Search: Breadth First Search, Depth First Search, Depth Bounded DFS (DBDFS), Depth First Iterative Deepening (DFID), state space problems- water jug, missionaries and cannibals, Farmer-Goat-Cabbage problems. **Heuristic Search:** Heuristic Functions, Best First Search, Simple Hill Climbing, Steepest-Ascent Hill Climbing, Stochastic Hill Climbing. **Optimal Search:** A* algorithm, Iterative Deepening A*, Recursive Best First Search.	**12**
Unit - 2	**Problem Decomposition and Planning and Constraint Satisfaction Problem Decomposition:** Goal Trees, Rule Based Systems, Rule Based Expert Systems. **Planning:** STRIPS, Forward and Backward State Space Planning, Goal Stack Planning, Plan Space Planning. Constraint Satisfaction: N-Queens, Constraint Propagation. Game Playing: Minimax Algorithm, Alpha-Beta Pruning.	**11**

Unit - 3	**Knowledge Representation:** Propositions, First order Logic: Clausal form, Unification, Resolution. **Structured knowledge Representation:** Frames and semantic nets**.** **Expert System:** Architecture of Expert System, Role of Expert system in Knowledge acquisition.	**11**
Unit - 4	**No-Code and Low-Code AI Systems:**Introduction, Comparison of No-Code and Low-Code platforms, Core components of Low-Code platforms, Limitations and disadvantages, applications of Low-Code platforms. **Edge Computing:**Introduction, Need for Edge Computing, Use cases and benefits, Challenges, and future trends in Edge Computing. **Transfer Learning:**Introduction, Types of Transfer Learning, Pre-trained models and fine-tuning, Feature Extraction Techniques, applications, and Challenges and limitations of Transfer Learning.	**11**
Pedagogy:	Collaborative Learning, Problem-Solving	
References/ Readings:	**Text Books** 1. Deepak Khemani, A First Course in Artificial Intelligence, McGraw Hill Education, Sixth Edition, 2017 ISBN: 978-1259029981. 2. Reema Thareja, Artificial Intelligence: Beyond Classical AI, Pearson Education, First Edition, 2023, ISBN:978-9356069329. 3. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson, Fourth Edition, 2022, ISBN:978-9356063570. **Reference Books** 1. Elaine Rich, Kevin Knight and Nair, Artificial Intelligence, MedTech Science Press, Fourth Edition, 2024. 2. Nilsson Nils J, Artificial Intelligence: A Modern Approach, Pearson Education, Fourth Edition, 2022, ISBN:978-9356063570.	
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Apply suitable search methods to systematically explore state spaces to derive solutions to AI problems. CO2: Construct AI solutions by applying planning methods, rule-based approaches, CSP techniques, and game-playing strategies. CO3: Demonstrate concepts of expert systems and apply logical knowledge representation models to solve AI problems. CO4: Illustrate no-code/low-code AI systems, edge computing, and transfer learning concepts.	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-305**

Title of the Course : **Artificial Intelligence Lab**

Number of Credits : **1 P**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Python programming skills and basic knowledge of AI/ML concepts.	
Course Objectives:	**The course will enable students to:** 1. Apply search strategies to solve classical AI problems. 2. Implement heuristic algorithms for effective problem solving. 3. Develop simple AI applications using game-playing techniques. 4. Build basic knowledge-based systems and inference methods using Prolog.	
Content:	**List of Programs/Experiments**	**No of Hours**
	1. Program to implement any one classical AI problem: Water Jug / Missionaries and Cannibals / Farmer–Goat–Wolf problem. 2. Program to implement the Depth First Search (DFS) algorithm. 3. Program to implement the Breadth First Search (BFS) algorithm. 4. Program to implement the Best First Search algorithm. 5. Program to implement the Hill Climbing algorithm. 6. Program to implement a variation of the Hill Climbing algorithm. 7. Program to implement the A* search algorithm. 8. Program to implement a Game Playing algorithm for eight puzzle program/ Graph. 9. Program to develop an Expert System using Prolog. 10. Program to perform logical inference using Resolution in Prolog.	**30**
Pedagogy:	Inquiry based Learning, Constructive and Collaborative Learning.	
References/ Readings:	**Text Books** 1. Deepak Khemani, A First Course in Artificial Intelligence, McGraw Hill Education, Sixth Edition, 2017 ISBN: 978-1259029981. 2. Ela Kumar, Artificial Intelligence, TechSar Pvt. Ltd, 2013, ISBN: 978-8190656665.	

	3. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson, Third edition, 2003, ISBN :10: 0136042597. **Reference Books** 1. Elaine Rich, Kevin Knight and Nair, Artificial Intelligence, MedTech Science Press, Fourth Edition, 2024. ISBN:978-8196544577. 2. Nilsson Nils J, Artificial Intelligence: A Modern Approach, Pearson Education, Fourth Edition, 2022, ISBN:978-9356063570.
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Apply state-space search techniques to AI problems. CO2: Compare uninformed and informed search algorithms. CO3: Develop game-playing and expert system applications. CO4: Demonstrate logical inference using Resolution and Prolog.

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-306**

Title of the Course : **Design and Analysis of Algorithms**

Number of Credits : **2 (2L)**

Effective from AY : **2024-25**

Pre- requisites for the Course:	Basic General Programming Concepts	
Course Objectives:	**This course will enable students to:** 1. Understand fundamental algorithm concepts, performance analysis, divide-and-conquer strategies, and methods for solving recurrence relations. 2. Develop the ability to analyze and design algorithms using greedy and dynamic programming approaches for optimization and shortest-path problems. 3. Equip students with systematic problem-solving skills using backtracking and branch-and-bound techniques for combinatorial and optimization problems. 4. Introduce students to classical string-processing algorithms and foundational concepts of NP-Hardness, NP-Completeness.	
Units	**Contents:**	**No of Hours**
Unit-1	**Introduction** – Algorithm Specification, Performance Analysis – Space complexity, Time complexity, Asymptotic Notation. **Divide and Conquer** – General method, Binary search, Merge sort, Quick sort, Strassen's Matrix Multiplication; The substitution method for solving recurrences,The recursion-tree method for solving recurrences,The master method for solving recurrences	**8**
Unit-2	**The Greedy Method** – General method, Fractional Knapsack problem, Minimum cost spanning trees – Prims, Kruskal, Single source shortest path problem. **Dynamic Programming** – General Method, Multistage Graphs-Forward & Backward Approach, All pairs shortest Path, Single source shortest path	**7**
Unit-3	**Backtracking**: General Method, N-queens problem, Sum of subsets problem, graph colouring. **Branch and Bound**– 0/1 Knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution, Traveling salesperson problem.	**8**

Unit-4	**Internet Algorithms:** String and Pattern matching Algorithm– Brute Force, KMP, Boyer Moore, Text compression - Huffman Coding, Text similarity testing. **Randomised Algorithm** - Primality Testing. **NP-Hard and NP-Complete Problems** - Basic Concepts	**7**
Pedagogy:	**Blended learning approach** that includes hands-on practice, problem-solving, and real-world application development	
References/ Readings:	**Text Books** 1. E. Horowitz et al, Fundamentals of Computer Algorithms, 2nd Edition UP. ISBN: 9788173716126 2. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley. ISBN: 978-0-471-38365-9 3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press, 2009. ISBN: 9780262033848 **Reference Books:** 1. Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011. **ISBN:** 9780321573513	
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Analyze algorithm efficiency and employ divide-and-conquer techniques, including solving recurrence relations using standard methods. CO2: Apply greedy and dynamic programming methods to solve optimization and shortest-path problems. CO3: Evaluate combinatorial problems and select appropriate backtracking and branch-and-bound techniques for their solution.. CO4: Demonstrate classical string-matching and compression algorithms and explain the basics of NP-Hard and NP-Complete problems.	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-307**

Title of the Course : **Design and Analysis of Algorithms Lab**

Number of Credits : **2 P**

Effective from AY : **2024-25**

Pre- requisites for the Course:	Basic programming skills (C/C++/Java/Python) and understanding of data structures.	
Course Objectives:	**This course will enable students to:** 1. Apply and implement fundamental divide-and-conquer algorithms and analyze their empirical performance through experimentation. 2. Apply greedy and dynamic programming techniques to solve optimization and graph-related computational problems in a programming environment. 3. Develop backtracking and branch-and-bound algorithms to handle combinatorial and decision-making problems effectively. 4. Implement classical string-matching and text-compression algorithms.	
	Experiments No / List of Programs	**No of Hours**
Content:	1. Implement MinMax algorithm. 2. Implement Merge Sort and Quick Sort; measure performance on varying input sizes. 3. Implement Strassen's Matrix Multiplication and compare it with standard matrix multiplication. 4. Implement Fractional Knapsack problem using greedy technique. 5. Implement Prim's Minimum Cost Spanning Tree. 6. Implement Kruskal's Minimum Cost Spanning Tree. 7. Implement Dijkstra's Single Source Shortest Path algorithm. 8. Implement Dynamic Programming solutions for Multistage Graph. 9. Implement all pair shortest path using Dynamic programming. 10. Implement N-Queens problem using Backtracking. 11. Implement the Sum of Subsets problem using Backtracking. 12. Implement Graph Coloring using Backtracking. 13. Implement 0/1 Knapsack using Branch and Bound (LC strategy). 14. Implement string matching algorithms: (a)KMP (b)Boyer–Moore	**60**

	15. Implement Huffman Coding for text compression	
References/ Readings:	**Text Books** 1. E. Horowitz et al, Fundamentals of Computer Algorithms, 2nd Edition UP. ISBN: 9788173716126 2. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley. ISBN: 978-0-471-38365-9 3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press, 2009. ISBN: 9780262033848 **Reference Books:** 1. Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011. **ISBN:** 9780321573513	
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Apply divide-and-conquer algorithms. CO2: Analyze greedy and dynamic programming techniques to solve optimization and graph problems in a programming environment. CO3: Develop backtracking and branch-and-bound algorithms for solving combinatorial problems. CO4: Implement string-matching and compression algorithms and understand complexity through practical exercises.	

Professional Electives

Name of the Programme : B.E. Computer Engineering

Course Code : CMP-321

Title of the Course : Software Testing and Quality Assurance

Number of Credits : 3(3L)

Effective from AY : 2024-25

Pre- requisites for the Course:	Fundamentals of Software Engineering	
Course Objectives:	**This course will enable students to:** 1. Understand the importance of software systems meeting user expectations for quality using the ISO 25010 standard. 2. Recognize that software testing is the primary means to ensure software quality through verification and validation. 3. Plan and prepare alternatives for quality assurance, including static analysis, inspections, and defect prevention strategies. 4. Measure and analyze quality using reliability metrics and automation tools to close the feedback loop	
Contents:		**No. of Hours.**
UNIT- 1	Software Quality: Quality perspective and expectations; ISO/IEC 25010 Quality Model - Product Quality Attributes (Functional Suitability, Performance, Security, Reliability); Correctness and defects. Quality Assurance (QA): Classification; Defect prevention vs. Defect detection; QA activities in software processes (Agile vs Waterfall); Verification and validation perspective. Quality Management Systems: ISO 9000 series standards; Capability Maturity Model Integration (CMMI) levels and process improvement areas.	**12**
UNIT- 2	Testing Processes: Purposes, activities, and context; Functional vs. Structural testing; Agile Testing Principles (The Agile Testing Quadrants). Test Management: Test planning and preparation (Test Strategy vs Test Plan); Test execution and result measurement; Defect Management (Severity vs Priority, Bug Life Cycle). Coverage & Usage Testing: Checklist-based testing and limitations; Testing for partition coverage; Use Case Testing and User Story validation.	**11**

UNIT- 3	Input Domain & Boundary Testing: Equivalence Class Partitioning (ECP); Boundary Value Analysis (BVA); Simple domain analysis and extreme point combination strategies. Structural & Interaction Testing: Control Flow Testing (Statement, Branch, Path coverage); Data Flow Testing (Definition-Use pairs). Defect Prevention & Inspection: Root cause analysis (Pareto Charts, Fishbone); Software Inspection: Fagan Inspection (Process and Roles); Static Code Analysis (Automated Code Review tools like SonarQube)	**11**
UNIT- 4	Software Reliability: Basic ideas and concepts; Reliability Models (MTBF, Failure Intensity); Failure Containment vs. Fault Tolerance. Risk Identification: Basic ideas and concepts; Risk-Based Testing (RBT); Cost of Quality (CoQ) analysis. Software Testing Tools: Need for automated testing tools; The Test Automation Pyramid; Taxonomy of tools: Functional/Regression: Selenium WebDriver (Concepts), Performance: Apache JMeter (Concepts), Management: Jira/Bugzilla.	**11**
Pedagogy:	Learner centric teaching	
References/ Readings:	Textbooks: 1. Jeff Tian, Software Quality Engineering, Wiley, 2006. ISBN: 97804717134456 2. Naresh Chauhan, Software Testing: Principles and Practices, 2nd Edition, Oxford University Press, 2018. ISBN: 9780199465873 Reference Books: 1. Navneesh Garg, Test Automation using Selenium WebDriver, Adactin Group. ISBN **0992293510** 2. Paul Ammann and Jeff Offutt, Introduction to Software Testing, 2nd Edition, Cambridge University Press, 2016. ISBN : **9781107172012**	
Course Outcomes:	**Upon completion of the course, students will be able to:** CO1: Explain quantitative and technical methods to assure software quality using ISO 25010 and CMMI standards. CO2: Apply Black-box and White-box testing approaches to software modules. CO3: Analyze defect prevention techniques including Fagan Inspections and Static Code Analysis. CO4: Evaluate the selection of automated testing tools based on project requirements.	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-322**

Title of the Course : **Software Testing and Quality Assurance Lab**

Number of Credits :**1P**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Fundamentals of software engineering and basic object-oriented programming.	
Course Objectives:	**This course will enable students to:** 1. To translate functional requirements into executable test scenarios using manual black-box design strategies. 2. To implement automated test scripts for functional and regression testing using web automation frameworks. 3. To utilize industry-standard tools for defect management, static code analysis, and API validation. 4. To assess software scalability and reliability through performance testing methodologies under simulated load.	
Contents:	**List of Programs /Experiments**	**No of Hours**
	1. Create a Test Plan and Test Strategy document for a sample application. 2. Log and track defects using Bugzilla/Jira (Defect Lifecycle). 3. Design test cases for a system with complex rules using Decision Tables. 4. Perform code quality scans on a Java/Python project using SonarLint. 5. Write and execute Unit Tests using JUnit/PyTest. 6. Validate REST API endpoints (GET/POST) using Postman. 7. Automate simple form submissions using Selenium WebDriver. 8. Script to handle dynamic elements (Waits), Dropdowns, and Alerts using Selenium. 9. Automate tests using external data sources (Excel/CSV). 10. Create a Test Plan to simulate load using Apache JMeter	**30**
Pedagogy:	Active Learning, Reflective Learning	
References/ Readings:	Textbooks: 1. Jeff Tian, Software Quality Engineering, Wiley, 2006. ISBN: 97804717134456 2. Naresh Chauhan, Software Testing: Principles and Practices, 2nd	

	Edition, Oxford University Press, 2018. ISBN: 9780199465873 Reference Books: 1. Navneesh Garg, Test Automation using Selenium WebDriver, Adactin Group. ISBN **0992293510** 2. Paul Ammann and Jeff Offutt, Introduction to Software Testing, 2nd Edition, Cambridge University Press, 2016. ISBN : **9781107172012**
Course Outcomes:	**After completing this course, students will be able to:** CO1: Apply defect tracking, static code analysis, and API validation tools to the software testing lifecycle. CO2: Develop automated test scripts for validating functional requirements using web automation frameworks. CO3: Evaluate system performance metrics including throughput and response time under simulated load conditions. CO4: Design comprehensive test scenarios using equivalence partitioning, boundary value analysis, and decision tables.

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-323**

Title of the Course : **Blockchain Technology**

Number of Credits : **3(3L)**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Expertise in programming, basic knowledge of computer security, networking and number theory.	
Course Objectives:	**This course will enable students to:** 1. Understand the building blocks of Blockchain. 2. Explore Cryptocurrency and Smart Contract 3. Study Private and Consortium Blockchain Systems 4. Analyze the limitations/challenges, security aspects and applications of Blockchain	
Contents:		**No of Hours**
Unit-1	**Fundamentals of Blockchain:** Introduction, of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain, The technology and the future. **Blockchain types and Consensus Mechanism:** Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol	**12**
Unit 2	**Cryptocurrency – Bitcoin, Altcoin and Token:** Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrency, Cryptocurrency Usage. **Smart Contracts:** Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum	**11**
Unit 3	**Private Blockchain System:** Introduction, Key Characteristics of Private Blockchain, Why We Need Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, Smart Contract in Private Environment. **Consortium Blockchain:** Introduction, Key Characteristics of Consortium Blockchain, Why We Need Consortium Blockchain.	**11**
Unit 4	**Security in Blockchain:**	**11**

	Security Aspects in Bitcoin, Security and Privacy Challenges of Blockchain in General, Performance and Scalability, Identity Management and Authentication **Limitations and Challenges of Blockchain**: Blockchain Implementation – Limitations, Challenges. **Applications of Blockchain : Education, Real-Estate**	
Pedagogy:	Inquiry based learning, Integrative, Reflective Learning , Constructive learning and Collaborative learning.	
Readings	**Text Books:** Chandramouli Subramanian, Asha A George , Abhilash K A , Meena Karthikeyan, Blockchain Technology, Universities Press (India) Pvt. Ltd., Edition Publication,2020,H. S. Kalsi; Electronic Instrumentation; Tata McGraw Hill. ISBN:9789389211634 **Reference Books:** 1. Dhillon, V., Metcalf, D., and Hooper, M, Blockchain enabled applications, 2017, 1st Edition, CA: Apress, Berkeley. ISBN:9781484230817 2. Wattenhofer, R. P, Distributed Ledger Technology: The Science of the Blockchain (Inverted Forest Publishing), 2017, 2nd Edition, Createspace Independent Pub, Scotts Valley, California, US.Newnes , An imprint of Elsevier.ISBN: 978-1544232102	
Course Outcomes	**After completing this course, students will be able to:** CO1: Understand the blockchain ecosystem and its real-world services. CO2: Explain the basics of cryptocurrency and smart contracts. CO3: Analyze private and consortium blockchain systems. CO4: Evaluate security aspects, limitations, challenges, and applications of blockchain technology.	

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-324**

Title of the Course : **Blockchain Technology Lab**

Number of Credits : **1P**

Effective from AY : **2024-25**

Pre-requisites for the Course:	**C++, Python, Javascript**	
Course Objectives:	**This course will enable students to:** 1. Understanding the basic Blockchain ecosystem and its applications 2. Explain wallets and perform transactions using Metamask,JSDK and Python 3. Create smart contracts using Solidity 4. Evaluate the application of Blockchain in real world scenarios	
Contents:		**No of Hours**
Experiment List	1. Case study on Ethereum Virtual Machine (EVM) and explain the core terminology used in the Blockchain ecosystem 2. Installing and Setting Up MetaMask , Creation of wallets using MetaMask, and Send/Receive tokens in Metamask using a Test Network 3. Create and run the "HelloWorld" Solidity program using Remix 4. Develop a basic token sale contract and voting contract using Solidity 5. Develop a NFT Marketplace contract using Solidity 6. Blockchain in financial services/retail/healthcare - case study 7. Creation of wallet using python 8. Set up Java SDK blockchain Network 9. Connecting Remix with Metamask 10. Mini Project	**30**
Pedagogy:	Instructional and Collaborative Learning, Inquiry Based Learning, Experiential Learning.	
Readings	**Text Books:** 1. Chandramouli Subramanian, Asha A George , Abhilash K A , Meena Karthikeyan, Blockchain Technology, Universities Press (India) Pvt. Ltd., Edition Publication,2020H. S. Kalsi; Electronic Instrumentation; Tata McGraw Hill.ISBN:9789389211634 2. Rendezvous with Practical Solidity : A COMPLETE HANDS ON	

	BOOK, Notion Press January 31, 2020 , ISBN: 9781647606626 **Web References**: 1. https://docs.metamask.io/ 2. Solidity — Solidity 0.8.31-develop documentation
Course Outcomes	**After completing this course, students will be able to:** CO1: Understand the Blockchain architecture and its applications CO2: Explain wallets and perform transactions using Metamask,JSDK and Python CO3: Create smart contracts using Solidity CO4: Evaluate the application of Blockchain in real world scenarios

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-325**

Title of the Course : **Soft Computing**

Number of Credits : **3(3L)**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Foundation in mathematics, specifically linear algebra, calculus, set theory	
Course Objectives:	**This course will enable students:** 1. Understand the fundamental concepts soft computing techniques and their applications. 2. Explain neural network learning algorithms and fuzzy logic principles. 3. Describe the working principles of Genetic Algorithms and other nature-inspired optimization. 4. Apply hybrid soft computing techniques such as Neuro-Fuzzy and Neuro-Genetic systems.	
Contents:		**No of Hours**
Unit-1	**Soft Computing:** Introduction to soft computing, soft computing vs. hard computing, various types of soft computing techniques, applications of soft computing. **Neural Networks:** Mc-Culloch Pitt's neuron model, Activation functions, Basic gates, Neural learning. Training algorithms-Hebbian learning rule, perceptron learning rule, Delta learning rule, Widrow-Hoff learning rule related problems. Error back propagation algorithm or generalized delta rule. Setting of parameter values and design considerations (Initialization of weights, Frequency of weight updates, Choice of learning rate, Momentum, Generalizability, Network size, Sample size, Non-numeric inputs).	**12**
Unit - 2	**Fuzzy Logic:** Introduction, Classical Set Theory (Crisp Set): Operations & Properties, Fuzzy Set Theory: Operations & Properties, Membership Functions and types, Fuzzy v/s Crisp Sets, Classical relations (Cartesian product) and Fuzzy relations: Cardinality, Operations, Properties and Composition, Tolerance and Equivalence Relations. Crisp Logic v/s Fuzzy logic, Fuzzy logic operations: AND, OR, NOT, Implication, Aggregation and Deffuzification, Lambda-cuts or Alpha-cuts for fuzzy, Types of defuzzification. Fuzzy Inference Systems and its design, Fuzzy Process, Type-2 fuzzy sets, Sugeno Fuzzy System.	**11**

Unit - 3	**Evolutionary algorithms:** Genetic algorithms -Fundamentals, basic concepts, working principle, encoding, fitness function, reproduction. Genetic modeling - Inheritance operator, cross over, inversion & deletion, mutation operator, Bitwise operator ,Generational Cycle, Convergence of GA, Applications & advances in GA, Differences & similarities between GA & other traditional methods. **Other Optimization Techniques:** Particle Swarm Optimization, Ant Colony Optimization, Travelling Salesman Problem.	**11**
Unit - 4	**Deep Neural Networks:** Introduction & Necessity of deep neural networks (DNN), Example: Auto encoder DNN, Convolutional neural networks: Convolution operation, Motivation and Pooling. **Hybrid Systems:** Sequential, Auxiliary and Embedded Hybrid Systems, Neuro-Fuzzy Hybrid System: Comparison, Characteristics & Classification, Neuro-Genetic Hybrid: Properties, GA based Back Propagation Network and its advantages, Genetic- Fuzzy and Fuzzy-Genetic Hybrid systems: Tuning, Learning, Advantages.	**11**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Text Books** 1. Anupam Shukla, Ritu Tiwari, Rahul Kala, Real Life Applications of Soft Computing, CRC Press, 2010, ISBN: 978-1-4398-2289-0 2. S. Rajasekaran & G.A. Vijayalakshmi Pai, Neural Networks, Fuzzy systems & Evolutionary Algorithms, Synthesis & applications, 2nd Edition, PHI Publication, 2017, ISBN: 9788120353343. **Reference Books** 1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016, MIT Press, ISBN: 9780262035613. 2. Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka, Elements of Artificial Neural Network, MIT press, 2000, ISBN: 0-262-13328-8. 3. J. Zurada, Introduction to Artificial Neural Networks, 2nd Edition, Jaico Publishing House, 2004, ISBN: 9788172246501. 4. S.N. Sivanandam& S.N. Deepa, Principles of Soft Computing, Wiley Publications, 2nd Edition, 2011, ISBN: 9788126527410.	
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Identify various soft computing techniques and illustrate their real-time applications. CO2: Demonstrate neural network learning methods and interpret fuzzy logic operations. CO3. Analyze various evolutionary algorithms.	

	CO4: Implement hybrid soft computing models such as Neuro-Fuzzy and Neuro-Genetic systems.

Name of the Programme : **B.E. Computer Engineering**

Course Code : **CMP-326**

Title of the Course : **Soft Computing Lab**

Number of Credits : **1(2P)**

Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of programming	
Course Objectives:	**The course will enable students to:** 1. Understand the fundamental concepts soft computing techniques and their applications. 2. Explain neural network learning algorithms and fuzzy logic principles. 3. Analyze the working principles of Genetic Algorithms and other nature-inspired optimization. 4. Integrate hybrid soft computing techniques for classification and optimization.	
Contents	**List of programs**	**No of Hours**
	1. Implementation of Simple Neural Network (McCulloh-Pitts model) 2. Implementation to find weight matrix and bias of HebbNet. 3. Implementation of Perceptron net for basic logic gates. 4. Implement Back-propagation algorithm. 5. Implement operations on fuzzy sets. 6. Design fuzzy inference system for a given problem. 7. Implementation of Genetic Algorithms to solve an optimization problem. 8. Implementation of Particle Swarm Optimization to solve optimization problems. 9. Implement Travelling salesman problem using Genetic Algorithm. 10. Implement CNN classifier.	**30**
Pedagogy:	Integration of instructional learning, constructive thinking, inquiry based, collaborative, experimental, and problem-solving approaches.	
References/ Readings:	**Text Books** 1. Anupam Shukla, Ritu Tiwari, Rahul Kala, Real Life Applications of Soft Computing, CRC Press, 2010, ISBN: 978-1-4398-2289-0 2. S. Rajasekaran & G.A. Vijayalakshmi Pai, Neural Networks, Fuzzy systems & Evolutionary Algorithms, Synthesis & applications, 2nd	

	Edition, PHI Publication, 2017, ISBN: 9788120353343. **Reference Books** 1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016, MIT Press, ISBN: 9780262035613. 2. Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka, Elements of Artificial Neural Network, MIT press, 2000, ISBN: 0-262-13328-8. 3. J. Zurada, Introduction to Artificial Neural Networks, 2nd Edition, Jaico Publishing House, 2004, ISBN: 9788172246501. 4. S.N. Sivanandam& S.N. Deepa, Principles of Soft Computing, Wiley Publications, 2nd Edition, 2011, ISBN: 9788126527410.
Course Outcomes:	**Upon completion of the course, students will be able to**: CO1: Implement Artificial Neural Network models including Perceptron, back propagation, and simple logic function networks. CO2: Construct fuzzy sets and design fuzzy inference systems. CO3: Utilize Genetic Algorithms and other optimization algorithms for engineering optimization problems. CO4: Evaluate and compare the performance of hybrid soft computing systems.

Semester VI

Major Courses

Name of the Programme : B.E. in Computer Engineering
Course code : CMP-308
Title of the course : Virtualization and Cloud Computing
Number of Credits : 3 (3L)
Effective from AY : 2024-25

Pre- requisites for the Course:	Knowledge of Fundamentals of Computers and Networking.	
Course Objectives:	The course will enable students to: 1. To introduce the fundamentals of virtualization and cloud computing, 2. To provide knowledge of cloud platforms, tools and services. 3. To explore commercial cloud systems, applications and deployment. 4. To build a foundation for advanced study, research and development of cloud-based solutions.	
Units	**Contents:**	**No of Hours**
Unit-1	Cloud Computing Fundamental: Motivation for Cloud Computing, Defining Cloud Computing, 5-4-3 Principles of Cloud computing, Cloud Ecosystem, Requirements for Cloud Services, Benefits and Drawbacks. Cloud Computing Architecture and Management: Introduction, Cloud Architecture, Anatomy of the Cloud, Applications on the Cloud, Managing the Cloud, Migrating Application to Cloud.	**11**
Unit-2	Cloud Deployment Models: Introduction, Private Cloud, Public Cloud, Community Cloud and Hybrid cloud. Cloud Service Models: Introduction, IaaS, PaaS, SaaS. Virtualization: Introduction, Virtualization Opportunities, Approaches to Virtualization, Hypervisors, Types of Hypervisors, Security Issues and Recommendations.	**11**
Unit-3	Technological Drivers for Cloud Computing: Introduction, SOA and Cloud, Services architectural model of SOA, Benefits of SOA. Cloud Service Providers: An Overview, Tools for IaaS: Amazon Web Services, Open Nebula, OpenStack, Eucalyptus, Tools for PaaS: Windows Azure, Google App Engine. Tools for SaaS: Apache VCL, Salesforce.com.	**11**
Unit-4	Introduction to Containers (Docker) and Orchestration (Kubernetes – overview)	**12**

	Serverless Computing- Definition & characteristics of serverless/cloud functions, Function-as-a-Service (FaaS) model, BaaS, Comparison with traditional VM and container-based deployment, Cloud vs Edge vs Fog computing.	
Pedagogy:	Interactive, reflective and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books:** 1. Rajkumar Buyya, Christian Vecciola , Thamarai Selvi , Shivananda Poojara , Satish N. Srirama, Mastering Cloud Computing- Powering AI, Big Data and IOT Applications, Mc. Graw Hill , Second Edition 2024, ISBN: 9355329504. 2. K. Chandrasekaran, Essentials of Cloud Computing, First Edition, Chapman and Hall/CRC, 2014, ISBN:1482205432. **Reference Books:** 1. Matthew Portnoy, Virtualization Essentials, Wiley First Edition, 2023, ISBN: 1394181566. 2. Gautam Shroff, Enterprise cloud computing, Technology Architecture, Applications, Cambridge University Press, First Edition 2010, ISBN: 1444192035.	
Course Outcomes:	**Upon successful completion of this course, students will be able to:** CO 1: Compare various cloud computing platforms and virtualization technologies. CO 2: Analyze performance, scalability and availability of cloud and virtualized environments. CO 3: Design and implement solutions to real-world problems using virtualization and cloud computing. CO 4: Evaluate commercial cloud services and emerging trends.	

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-309**
Title of the course : **Virtualization and Cloud Computing Lab**
Number of Credits : **1(1P)**
Effective from AY : **2024-25**

Prerequisites for the Course:	Knowledge of Programming like C, C++, Python and Basic Mathematics.	
Course Objectives:	The course will enable students to: 1. Apply fundamental concepts of cloud computing and virtualization to analyze different cloud platforms 2. Demonstrate practical skills in deploying and managing cloud-based services through hands-on experiments and real-world case studies. 3. Utilize cloud tools, technologies and cloud automation mechanisms. 4. Evaluate enterprise cloud strategies based on industry case studies.	
Content:	List of Programs /Experiments	**No of Hours**
	1. Study and deployment of IaaS service delivery models	**30**
	2. Study and deployment of PaaS service delivery models	
	3. Study and deployment of SaaS service delivery models	
	4. Study of various virtualization techniques in cloud	
	5. Study and deployment various cloud deployment models	
	6. Study of various Service Oriented Architectures (XML, SOAP, WSDL, UDDI)	
	7. Study and deployment of Google and Amazon Services	
	8. Study and deployment of Microsoft and Salesforce Services	
	9. Study and deployment of Docker / Kubernetes architectures.	
	10. Study and analysis of Serverless/Edge/Fog computing	
Pedagogy:	Interactive, reflective and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books:** 1. Rajkumar Buyya, Christian Vecciola , Thamarai Selvi , Shivananda Poojara , Satish N. Srirama, Mastering Cloud Computing- Powering AI, Big Data and IOT Applications, Mc. Graw Hill , Second Edition 2024, ISBN: 9355329504. 2. K. Chandrasekaran, Essentials of Cloud Computing, First Edition, Chapman and Hall/CRC, 2014, ISBN:1482205432.	

	Reference Books: 1. Matthew Portnoy, Virtualization Essentials, Wiley First Edition, 2023, ISBN: 1394181566. 2. Gautam Shroff, Enterprise cloud computing, Technology Architecture, Applications, Cambridge University Press, First Edition 2010, ISBN: 1444192035.
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Deploy and implement various service delivery models. CO 2. Install Virtual machines using virtualization technologies. CO 3. Demonstrate the applications of Cloud Computing for Business. CO 4. Apply the skills and knowledge learnt in cloud computing and virtualization to incorporate in emerging trends like fog/edge/serverless computing.

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-310**
Title of the course : **Operating Systems**
Number of Credits : **3 (3L)**
Effective from AY : **2024-25**

Pre- requisites for the Course:	Knowledge of Fundamentals of Computers.	
Course Objectives:	The subject aims to provide the student with: 1. An understanding of the fundamental concepts, structure, services, and system calls of an operating system and its role in managing computer hardware and software resources. 2. An understanding of process management, multithreading models, CPU scheduling algorithms, and real-time scheduling techniques used in modern operating systems. 3. The ability to analyse and understand synchronization mechanisms, deadlocks, and memory management techniques including virtual memory and page replacement algorithms. 4. An understanding of file system implementation, I/O systems, secondary storage management, and necessary skills required for Shell Programming.	
Units	**Contents:**	**No of Hours**
Unit-1	**Introduction to Operating Systems:** Abstract view of a Computer System, What Operating Systems do, Computer System Architecture, Operating System Operations, Operating System Services, System calls, Types of System calls. **Process management:** Processes concept, Process scheduling, Operations on processes, Inter-process communication. **Threads &Concurrency :** Overview, Multithreading models, Threading issues. **CPU Scheduling**: Basic Concepts, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF / SRTN, Priority Scheduling, Round Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Multiprocessor Scheduling, Real Time Scheduling: RM, EDF	**12**
Unit-2	**Synchronization Tools and Examples**: Critical Section Problem, Petersons solution, Synchronization hardware support, Mutex locks, Semaphores, Classical problems of synchronization using semaphores (Producer – Consumer problem, Readers – Writers problem, Dining philosophers Problem), Monitors (Dining philosophers Problem).	**11**

	Deadlocks: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.	
Unit-3	**Memory Management:** Background, Contiguous allocation, Segmentation, Paging, Structure of the page table **Virtual Memory**: Demand Paging, Page replacement algorithms (FIFO, Optimal page replacement, Least Recently used), Allocation of frames, Thrashing. **File System Interface**: File Concept, Access methods, Directory Structure. **File system implementation**: Allocation methods	**11**
Unit-4	**I/O Systems:** I/O Hardware, Application I/O Interface, Kernel I/O subsystem. **Secondary Storage structure**: Hard Disk Drives, HDD scheduling, Storage Device management **Linux Commands**: Basic Linux commands, Essential Shell Programming.	**11**
Pedagogy:	Interactive, reflective and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books:** 1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; Operating System Concepts; 10th Edition; Wiley; 2018. ISBN: 1119800366. 2. Sumitabha Das; UNIX – Concepts and applications; 4th edition; McGraw Hill Education; 2017, ISBN: 0070635463. **Reference Books:** 1. William Stallings; Operating systems- Internals and design principles; 9th edition; Pearson, 2018, ISBN: 0134670957. 2. A.S Tanenbaum, Albert Woodhull; Operating systems- Design and implementation; 3rd edition; Pearson; 2015, ISBN: 0131429388. 3. Milan Milenkovic; Operating Systems, 2nd edition, Tata McGraw Hill; 2001, ISBN: 0074632728.	
Course Outcomes:	**Upon successful completion of this course, students will be able to:** CO1. Explain the architecture, structure, and services of operating systems, including system calls and process life-cycle management. CO2. Apply CPU scheduling, process synchronization, and deadlock handling algorithms to solve operating system problems. CO3. Analyze memory management schemes, virtual memory techniques, and page replacement algorithms. CO4. Demonstrate understanding of file systems, I/O management, disk scheduling, and perform basic Linux commands and shell scripting tasks.	

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-311**
Title of the course : **Operating Systems Lab**
Number of Credits : **1P**
Effective from AY : **2024-25**

Pre- requisites for the Course:	Knowledge of Fundamentals of Computers.	
Course Objectives:	The subject aims to provide the student with: 1. A comprehensive understanding of the underlying principles, techniques and approaches in operating systems. 2. Necessary skills required to implement process management, threads, CPU scheduling and synchronization mechanisms. 3. Necessary skills required to implement memory management and deadlock avoidance techniques. 4. Necessary skills required for Shell Programming.	
	List of Experiments:	**No of Hours**
Content:	1. Process creation using system calls 2. Non-preemptive CPU scheduling algorithms 3. Preemptive CPU scheduling algorithms 4. Implementation of threads 5. Process synchronization using semaphores 6. Implementation of deadlock avoidance scheme 7. Implementation of Paging/ Segmentation Page replacement methods 8. Implementation of memory allocation methods 9. Implementation of Page replacement schemes 10. Implementation of Disk scheduling algorithms	**30**
Pedagogy:	Interactive, reflective and collaborative methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books:** 1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; Operating System Concepts; 10th Edition; Wiley; 2018. ISBN: 1119800366. 2. Sumitabha Das; UNIX – Concepts and applications; 4th edition; McGraw Hill Education; 2017, ISBN: 0070635463. **Reference Books:** 1. William Stallings; Operating systems- Internals and design principles; 9th edition; Pearson, 2018, ISBN: 0134670957. 2. A.S Tanenbaum, Albert Woodhull; Operating systems- Design and implementation; 3rd edition; Pearson; 2015, ISBN: 0131429388.	

	3. Milan Milenkovic; Operating Systems, 2nd edition, Tata McGraw Hill; 2001, ISBN: 0074632728..
Course Outcomes:	**Upon successful completion of this course, students will be able to:** CO1. Write programs for process and thread management using system calls. CO2. Implement CPU scheduling algorithms. CO3. Run programs on process synchronization mechanisms, deadlock avoidance techniques and memory management techniques. CO4. Develop shell scripts for simple applications and execute various UNIX commands on a standard UNIX/LINUX operating system.

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-312**
Title of the course : **Machine Learning**
Number of Credits : **2 (2L)**
Effective from AY : **2024-25**

Pre- requisites for the Course:	Basic understanding of probability, statistics, and linear algebra.	
Course Objectives:	This course will enable students to : 1. Explain the fundamental concepts of machine learning, including learning paradigms, concept learning, model selection, and evaluation. 2. Apply supervised learning techniques to solve real-world prediction and classification problems. 3. Implement unsupervised learning methods for pattern discovery and data analysis. 4. Use feature engineering, dimensionality reduction techniques, and reinforcement learning concepts to build efficient machine learning systems.	
Units	**Contents:**	**No of Hours**
Unit-1	Introduction to Machine Learning: Need for Machine learning, types of Machine Learning, Challenges of Machine Learning, and Machine Learning process. Concept Learning and the General-to-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search: General to specific ordering of Hypothesis, FIND-S, Version Spaces and the CANDIDATE-ELIMINATION Algorithm. Induction Biases: Bias and Variance, Bias vs. Variance Trade-off , Best Fit in Machine Learning , Modelling in Machine Learning : Model Selection and Model Evaluation, Re-sampling Methods	**8**
Unit-2	Supervised Learning Similarity-based learning: Introduction, K-Nearest Neighbour Regression Analysis: Linear Regression, Logistic regression Decision Trees Learning: ID3 Algorithm, Pruning of Decision trees Bayesian Learning-Probability-based Learning: Fundamentals of Bayes Theorem, Naïve Bayes Algorithm Ensemble Learning: Random Forest, Support Vector Machine Algorithm	**7**
Unit-3	Un-Supervised Learning Hierarchical Clustering: single linkage, complete linkage, average linkage.	**8**

	Partition Clustering: K Means Clustering Density-based spatial clustering Probability Model based Methods: Fuzzy clustering, Expectation Maximization algorithm.	
Unit-4	Feature Engineering and Dimensionality Reduction: What is Feature Engineering, Principal Component Analysis, Linear Discriminant Analysis, and Singular Value Decomposition. Reinforcement Learning: Overview, Components of Reinforcement Learning, Q-Learning	**7**
Pedagogy:	Blended learning approach that includes hands-on practice, problem solving, and real-world application development	
References/ Readings:	**Text Books** 1. Mitchell, Tom. Machine Learning. McGraw-Hill, 2013, ISBN: 0070428077. 2. Sridhar, S., and M. Vijayalakshmi. Machine Learning. Oxford University Press, 2021, ISBN: 0190127279. **Reference Books:** 1. Alpaydın, Ethem. Introduction to Machine Learning. Fourth ed., MIT Press, 2020, ISBN: 0262043793. 2. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006, ISBN: 1493938436.	
Course Outcomes:	After going through this course, the students will be able to: CO1: Explain key machine learning concepts, learning paradigms, concept learning, and model evaluation. CO2: Describe supervised learning methods including similarity-based approaches, regression, decision trees, Bayesian learning, ensembles, and SVMs. CO3: Distinguish among unsupervised learning techniques such as hierarchical, partition-based, density-based, fuzzy, and probabilistic clustering. CO4: Discuss feature engineering, dimensionality reduction methods, and basic reinforcement learning principles.	

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-313**
Title of the course : **Machine Learning Lab**
Number of Credits : **2P**
Effective from AY : **2024-25**

Pre- requisites for the Course:	Python programming and basics of probability, statistics, and linear algebra.	
Course Objectives:	**This course will enable students to:** 1. Develop practical skills to implement fundamental machine learning algorithms for classification, regression, and clustering. 2. Apply model evaluation techniques and compare the performance of various ML models. 3. Implement advanced machine learning methods including probabilistic models, ensemble techniques, and dimensionality reduction approaches. 4. Build and analyze basic reinforcement learning algorithms through hands-on experimentation.	
Units	**List of Programs /Experiments No. of Hours**	**No of Hours**
Content:	**List of Experiments:** 1. Implementation of concept learning using FIND-S and Candidate-Elimination algorithms. 2. Implement different model evaluation techniques. 3. Implement Linear Regression for prediction tasks. 4. Implement Logistic Regression for binary classification. 5. Implement Decision Tree using the ID3 algorithm. 6. Implement a Naïve Bayes classifier. 7. Implement Support Vector Machine for classification. 8. Implement a Random Forest classifier. 9. Implement K-Means clustering. 10. Implement Hierarchical Clustering. 11. Implement Density-Based Spatial Clustering. 12. Implement Clustering using Fuzzy C-Means and Gaussian Mixture Models (GMM). 13. Implement Principal Component Analysis. 14. Implement Q-Learning reinforcement learning algorithm. 15. Case study on Scikit-learn python library.	**60**
Instructions:	Minimum 15 experiments to be performed.	

References/ Readings:	**Text Books** 1. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 3rd ed., O'Reilly Media, 2022, ISBN: 9355421982 2. Müller, Andreas C., and Sarah Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists. 1st ed., O'Reilly Media, 2016, ISBN: 9352134575. **Reference Books:** 1. Alpaydın, Ethem. Introduction to Machine Learning. Fourth ed., MIT Press, 2020, ISBN: 0262043793. 2. Mitchell, Tom. Machine Learning. McGraw-Hill, 2013, ISBN: 0070428077. 3. Sridhar, S., and M. Vijayalakshmi. Machine Learning. Oxford University Press, 2021, ISBN: 0190127279.
Course Outcomes:	**After going through this course, the students will be able to:** CO1: Implement fundamental ML algorithms such as FIND-S, Candidate Elimination, Linear/Logistic Regression, Decision Trees, Naïve Bayes, and SVM. CO2: Apply model evaluation methods and ensemble techniques like Random Forest to analyze ML model performance. CO3: Execute unsupervised learning algorithms including K-Means, Hierarchical Clustering, DBSCAN, Fuzzy C-Means, and GMM using EM. CO4: Develop dimensionality reduction and basic reinforcement learning (QLearning) techniques in practical scenarios and apply the foundational library for classical machine learning in Python.

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-314**
Title of the course : **Image Processing**
Number of Credits : **3 (3L)**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of Mathematics, Signals and systems, and Digital processing.	
Course Objectives:	**This course will enable students:** 1. Introduce the fundamental concepts of digital image processing including image formation, pixel relationships, sampling, quantization, and basic intensity transformations. 2. Develop the ability to apply spatial and frequency domain filtering techniques for image enhancement, smoothing, sharpening, and restoration. 3. Enable understanding and implementation of advanced image processing methods including color image processing, morphological operations, segmentation techniques, and feature representation. 4. Familiarize students with image degradation models, compression techniques, and feature extraction methods used in real-world image analysis and computer vision applications.	
Contents:		**No of Hours**
Unit-1	**Digital Image Fundamentals:** – What is Digital Image Processing?, Examples of fields that use DIP, Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Brightness Adaptation and discrimination, Image Sensing and Acquisition, Image Sampling and Quantisation, Some Basic Relationships Between Pixels **Intensity Transformation –** Background, Some basic intensity Transformation functions, Histogram Processing - Histogram Equalization, Histogram Matching.	**11**
Unit - 2	**Spatial Filtering –** Fundamentals of spatial Filtering, Smoothing Spatial Filtering, Sharpening Spatial Filtering, **Filtering in the Frequency Domain -** Introduction, 2D Discrete Fourier Transform (DFT), Properties of Fourier transforms, The Basics of Filtering in the Frequency Domain, Image Smoothing and Sharpening Using Frequency Domain – Ideal, Butterworth Gaussian Filters. **Color Image Processing-** Color Fundamentals, Color models, Pseudocolor Image processing	**11**
Unit - 3	**Image Restoration** - A Model of the Image Degradation/Restoration Process, Minimum Mean Square Error	**12**

	(Wiener) Filtering. **Image Compression** - Fundamentals, Image Compression Models, Some Basic Compression Methods - Huffman Coding, LZW coding, Block Transform coding, JPEG Coding. **Image Segmentation**: Fundamentals, Basic edge Detection, Thresholding- Foundation, Basic Global Thresholding, Thresholding using Otsu Method, canny edge Detection method Region-based Segmentation, Watershed Segmentation Algorithm.	
Unit - 4	**Morphological Image Processing**: Preliminaries, Erosion and Dilation, Opening and Closing, The Hit-or-Miss Transformation, Some Basic Morphological Algorithms – Boundary Extraction, Hole Filling, Convex hull, Thinning, Skeletons **Representation and Description** Representation – Boundary following, chain codes, Boundary Descriptors - Simple Descriptors, shape numbers, Regional Descriptors - Simple Descriptors, Topological Descriptors	**11**
Pedagogy:	Reflective, Inquiry based, Problem based, Blended learning.	
References/ Readings:	**Text Books** 1. Meiqing Wang, Choi-Hong Lai, A Concise Introduction to Image Processing Using C++, First Edition, CRC Press, 2008, ISBN: 1138051640. 2. R.C. Gonzalez and R.E. Woods, Digital Image Processing Fourth Edition, Addison Wesley, 2019 ISBN: 9789353062989 **Reference Books** 1. Anil K. Jain, Fundamentals of Digital Image Processing, First Edition, Pearson Education, 2015. ISBN: 933255191X 2. Madhuri A. Joshi, Digital Image Processing - An Algorithmic Approach, Second Edition, PHI, 2018. ISBN: 9387472582 3. William K.Pratt, Digital Image Processing, Fourth Edition,John-Wiley & Sons, 2006. ISBN: 9788126526840 4. Milan Sonka, Roger Boyle&VaclavHlavac, Digital Image Processing and Computer Vision, First Edition, Cengage Learning India, 2008. ISBN: 8131505553.	
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain the fundamental concepts of digital image processing, including image acquisition, pixel relationships, sampling, quantization, intensity Transformations, and histogram processing. CO 2. Apply spatial and frequency domain filtering techniques such as smoothing, sharpening, Fourier-based filtering, and color processing to enhance the image Quality.	

	CO 3. Compare image restoration methods, image compression techniques (Huffman, LZW, JPEG), and segmentation techniques. CO 4. Implement various morphological operations on the image.

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-315**
Title of the course : **Image Processing Lab**
Number of Credits : **1P**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of signal processing	
Course Objectives:	**This course will enable students:** 1. Understand the fundamental structure of digital images, pixel representation, and their spatial relationships. 2. Apply basic and advanced image enhancement techniques, including spatial transformations, contrast adjustments, histogram processing, smoothing, and sharpening filters. 3. Implement core image processing techniques such as morphological operations, edge detection, segmentation, and compression for meaningful image analysis. 4. Gain hands-on proficiency in developing, testing, and evaluating image processing algorithms using computational tools.	
Sr.No	**List of Experiments**	**No of Hours**
1	Create and display an Image and basic relationships between pixels	**30**
2	Implementation of Basic Spatial Transformations of an Image	
3	Implement Contrast stretching on a low contrast image.	
4	Implement Histogram, and Histogram Equalization.	
5	Implementation of Image Smoothening Filters.	
6	Implementation of image sharpening filters.	
7	Implementation of image Compression techniques.	
8	Implementation of canny edge Detection method.	
9	Implement Watershed Segmentation Algorithm.	
10	Implementation of Morphological Image processing.	
Pedagogy:	Interactive, collaborative, and Demonstration-Based Learning, Experiential (Hands-on) Learning, Continuous Assessment & Reflection.	
References/ Readings:	**Text Books** 1. Meiqing Wang, Choi-Hong Lai, A Concise Introduction to Image Processing Using C++, First Edition, CRC Press, 2008, ISBN: 1138051640.	

	2. R.C. Gonzalez and R.E. Woods, Digital Image Processing Fourth Edition, Addison Wesley, 2019 ISBN: 9789353062989 **Reference Books** 1. Anil K. Jain, Fundamentals of Digital Image Processing, First Edition, Pearson Education, 2015. ISBN: 933255191X 2. Madhuri A. Joshi, Digital Image Processing - An Algorithmic Approach, Second Edition, PHI, 2018. ISBN: 9387472582 3. William K.Pratt, Digital Image Processing, Fourth Edition,John-Wiley & Sons, 2006. ISBN: 9788126526840 4. Milan Sonka, Roger Boyle&VaclavHlavac, Digital Image Processing and Computer Vision, First Edition, Cengage Learning India, 2008. ISBN: 8131505553..
Course Outcomes:	Upon completion of the course, students will be able to: CO1: Apply fundamental image processing operations such as image creation, display and analysis of pixel relationships. CO2: Implement key image enhancement techniques, including spatial transformations, Contrast stretching, histogram equalization, smoothing, and sharpening filters. CO 3: Apply morphological operations, edge detection (Canny), and compression methods for effective preprocessing and feature extraction. CO 4: Implement advanced image segmentation techniques such as Watershed to Partition images into meaningful regions for analysis.

Professional Electives

Name of the Programme : B.E. in Computer Engineering
Course code : CMP-327
Title of the course : Advanced Computer Graphics
Number of Credits : 3 (3L)
Effective from AY : 2024-25

Pre- requisites for the Course:	Knowledge of Basic Mathematics.	
Course Objectives:	This course will enable students to: 1. Knowledge about computer graphic hardware and software used. 2. Understanding of 2D and 3D graphics and their transformations. 3. Ability to appreciate the use of colour models. 4. Understanding of the methods used in modelling the motion in the virtual world.	
Units	**Contents:**	**No of Hours**
Unit-1	**Overview of graphic systems**: Raster scans systems, Random scan systems. **Output Primitives**: Points and lines, Line drawing algorithms, DDA, Bresenham's line algorithm, Circle generating algorithms, Properties of circles, Midpoint circle algorithm, Ellipse generating algorithm, Properties of Ellipses, Midpoint ellipse algorithm. **Filled area primitives**: Scan line polygon Fill algorithm, Inside – outside tests, Scan line fill of curved boundary, Boundary fill algorithm, Flood fill algorithm.	**11**
Unit-2	**Two Dimensional Geometric Transformations**: Basic Transformations, Translation, Rotation, Scaling, Composite transformation, Translations, Rotations, Scaling, Other transformations- Reflection, Shear. **Two-Dimensional Viewing**: The viewing pipeline, Viewing coordinate reference frame, Window to viewport coordinate transformation. **Clipping operations**: Point Clipping, Line clipping, Cohen-Sutherland Line Clipping, Polygon Clipping, Sutherland Hodgeman Polygon clipping, Weiler- Atherton Polygon Clipping, Curve clipping, Text clipping.	**12**
Unit-3	**Three Dimensional Geometric and Modeling transformations:** Translation Rotation, Coordinate Axes, rotations, Scaling, Reflections, Shears Three- Dimensional Viewing,	**11**

	Curves and Surfaces: Shape Description Requirements, Parametric Functions, Bezier Methods. B-Spline Methods.	
Unit-4	**Color Models and Color Applications**: Properties of light, Standard primaries and the, Chromaticity Diagram, XYZ Color model, CIE Chromaticity Diagram, RGB color model, YIQ Color Model, CMY Color Model, HSV Color Model, HLS Color Model. **Computer Animation**: Design of animation sequences, General computer animation functions, Raster Animations, Computer animation languages, Motion specification, Direct motion specification, Goal directed systems Kinematics and dynamics.	**11**
Pedagogy:	Interactive, reflective and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills	
References/ Readings:	**Text Books:** 1. Donald Hearn and M. P. Baker; Computer Graphics; Prentice Hall of India Pvt. Ltd. ISBN: 0131615300 2. William Newman and Robert Sproull; Principles of Interactive Graphics; Tata McGraw hill Publishing company Ltd. ISBN: 0070664552 **Reference Books:** 1. Foley, Van Dam, Feiner and Hughe; Computer Graphics: Principles and Practice, ISBN: 0321399528 2. N. Krishnamurthy; Introduction to Computer Graphics; TMH ISBN: 0070435367 3. Steven Harrington; Computer Graphics; Tata McGraw Hill. ISBN: 9339204808	
Course Outcomes:	**Upon successful completion of this course, students will be able to:** CO 1: Apply various graphics primitives to generate computer graphics. CO 2: Illustrate techniques of 2D transformations and clipping used in various graphic applications. CO 3: Analyze 3D transformations and represent curves along with their properties. CO 4: Demonstrate various color models and design of an animation sequence.	

Name of the Programme : **B.E. in Computer Engineering**
Course code : **CMP-328**
Title of the course : **Advanced Computer Graphics Lab**
Number of Credits : **1P**
Effective from AY : **2024-25**

<table>
<tr><td>Prerequisites for the Course:</td><td colspan="2">Knowledge of Programming like C, C++ and Basic Mathematics</td></tr>
<tr><td>Course Objectives:</td><td colspan="2">The subject aims to provide students with:
1. An introduction to the fundamental practical concepts of Computer Graphics.
2. Knowledge of computer graphics hardware and software.
3. The ability to develop effective programs to solve graphics programming challenges, including rendering different shapes.
4. Ability to implement methods used in modelling motion in the real world.</td></tr>
<tr><td rowspan="12">Content:</td><td>List of Programs /Experiments</td><td>No of Hours</td></tr>
<tr><td>1. Implement animation using inbuilt graphic functions.</td><td rowspan="11">30</td></tr>
<tr><td>2. Implement Line Drawing Algorithm.</td></tr>
<tr><td>3. Implement the Midpoint Circle Generation Algorithm.</td></tr>
<tr><td>4. Implement the Midpoint Ellipse Generation Algorithm.</td></tr>
<tr><td>5. Implement Boundary Fill Algorithm to fill a closed shape with color.</td></tr>
<tr><td>6. Implement Flood Fill Algorithm to fill a closed shape with color</td></tr>
<tr><td>7. Implement Basic 2D Transformations: Translation, Scaling, Rotation, Shearing and Reflection.</td></tr>
<tr><td>8. Implement Cohen-Sutherland Line Clipping Algorithm.</td></tr>
<tr><td>9. Implement Sutherland Hodgeman Polygon Clipping Algorithm.</td></tr>
<tr><td>10. To implement 3D animation using OpenGL, OpenCV.</td></tr>
<tr><td>Pedagogy:</td><td colspan="2">Interactive, reflective and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills</td></tr>
<tr><td>References/ Readings:</td><td colspan="2">Text Books:
1. Donald Hearn and M. P. Baker; Computer Graphics; Prentice Hall of India Pvt. Ltd. ISBN: 0131615300</td></tr>
</table>

	2. William Newman and Robert Sproull; Principles of Interactive Graphics; Tata McGraw hill Publishing company Ltd. ISBN: 0070664552 **Reference Books:** 1. Foley, Van Dam, Feiner and Hughe; Computer Graphics: Principles and Practice, ISBN: 0321399528 2. N. Krishnamurthy; Introduction to Computer Graphics; TMH ISBN: 0070435367 3. Steven Harrington; Computer Graphics; Tata McGraw Hill. ISBN: 9339204808
Course Outcomes:	**Upon completion of the course, students will be able to:** CO 1. Execute Line drawing algorithm, circle generating algorithm and ellipse generating algorithm. CO 2. Apply clipping and filling techniques for modifying an object. CO 3. Implement programs that demonstrate geometrical transformations. CO 4. Develop programs to create curves and animation.

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-329**
Title of the Course : **Software Defined Networks**
Number of Credits : **3 (3L)**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of Computer Networks (OSI, TCP/IP, Switching & Routing), Network Security fundamentals, Basic Linux commands and Virtualization concepts	
Course Objectives:	**The Course teachers are expected to impart the following:** 1. To develop students' ability to understand the concepts of traditional networks with its limitations and the need to move to Software Defined Networks. 2. To develop the students' ability to understand the fundamentals of SDN, its planar architecture and to understand the flexibility of multilevel pipeline processing 3. Critically analyze and appreciate the evolution of software defined networks 4. Design and develop various applications of SDN and Orchestrate different network services	
Contents:		**No of Hours**
Unit-1	**INTRODUCTION TO SDN** Overview, History, Evolution and Working of Software Defined Networks (SDN), Architecture of SDN, Modern Data Center – Traditional Switch Architecture, Centralized and Distributed Control and Data Planes.SDNFlavours, Scalability (Data Centers, Service provider networks, ISP Automation), Reliability (QoS, and Service Availability), Consistency (Configuration management, and Access Control Violations), Opportunities and Challenges. Applications of SDN.	**11**
Unit-2	**OPEN FLOW, SDN CONTROLLERS AND DATA CENTERS** Open Flow Specification, Drawbacks of Open SDN, SDN via APIs, SDN via Hypervisor Based Overlays, SDN via Opening up the Device, SDN Controllers, General Concepts, Multitenant and Virtualized Multitenant Data Center, SDN Solutions for the Data Center Network, VLANs, EVPN, VxLAN , NVGRE.	**11**
Unit - 3	**ARCHITECTURE AND FRAMEWORKS** Network Operating System (NOS), Languages and functions available for programming SDNs, northbound API. Mininet. Software vs. Hardware SDN switch implementations - Open vSwitch, WhiteBox, ONL. Controller implementations - POX, NOX, Beacon, Floodlight. Special Purpose controllers -	**11**

	Flowvisor, RouteFlow. Juniper SDN Framework, IETF SDN Framework, Open Daylight Controller, Floodlight Controller, Bandwidth Calendaring, DataCenter Orchestration.	
Unit-4	**PROGRAMMING AND TRAFFIC MANAGEMENT** Network Programmability, Network Function Virtualization, NetApp Development, Network Slicing, SDX; Current Languages and Tools, Composition of SDNs. Network Virtualization, Network Topology and Topological Information Abstraction, Data Centric Traffic Management, Wide Area Traffic Management, Wireless networks.	**12**
Pedagogy:	Instructional and Reflective Learning, Inquiry Based Learning, Experiential Learning.	
References/ Readings:	**Textbooks:** 1. Paul Goransson, Chuck Black, Timothy Culver, Software Defined Networks: A Comprehensive Approach (2nd Edition), Morgan Kaufmann,2016, ISBN: 978-0128045558 2. Rajendra Chayapathy et al, Network Function Virtualization (NFV) with a Touch of SDN, CRC Press,2020, ISBN: 978-0367352785 3. Thomas D. Nadeau, Ken Gray, SDN: Software Defined Networks – An Authoritative Review of Network Programmability Technologies, O'Reilly Media, 2013, ISBN: 978-1449342302 **Reference Books:** 1. Karthik Ramasamy, S. Mathew,OpenFlow Cookbook, Packt Publishing, 2016, ISBN: 978-1783983744. 2. Gerard Blanco, Radia Perlman, Interconnections of SDN and NFV, Addison-Wesley,2021, ISBN: 978-0139754851 3. SrilathaVemula, SaiMugunth , Software Defined Networking with ONOS, Packt Publishing, 2017, ISBN: 978-1786465988. **Supplementary Web references** 1. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf 2. http://mininet.org/ 3. https://www.opennetworking.org/	
Course Outcomes:	**Upon completion of the course, students will be able to:** CO1: Understand the evolution, concepts, architecture, and operation of Software Defined Networks. CO2: Analyze different SDN controllers, OpenFlow specifications, protocols, APIs, and virtualization techniques. CO3: Analyze the performance related problems of SDN, including those in routing, optimizing traffic engineering. CO4: Evaluate real-world SDN applications, scalability challenges, reliability mechanisms, and modern data center SDN solutions.	

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-330**
Title of the Course : **Software Defined Networks Lab**
Number of Credits : **1P**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Students should have basic knowledge of: • Basic understanding of Computer Networks concepts • Familiarity with Linux operating system and basic terminal commands • Fundamental knowledge of network virtualization tools or concepts • Basic programming skills in Python	
Course Objectives:	**The Course teachers are expected to impart the following:** 1. To introduce students to the fundamentals of Software Defined Networking through simulation using open-source tools such as Mininet, POX, RYU, and Open vSwitch. 2. To develop the ability to configure, analyze, and manage SDN components including controllers, switches, flow tables, and network topologies. 3. To provide hands-on experience in implementing OpenFlow rules, virtual network segmentation, traffic monitoring, and basic network programmability. 4. To enable students to design, simulate, and evaluate SDN-based network scenarios for understanding traffic behavior, controller performance, and programmable network operations.	
Content	**List of Experiments**	**No. of Hours**
	1. Simulation of a Basic Software-Defined Network Using Mininet 2. Demonstration of Control Plane–Data Plane Separation in an SDN Environment 3. Installation and Verification of OpenFlow Rules Using POX Controller 4. Simulation of VLAN-Based Network Segmentation Using Open vSwitch in Mininet 5. Comparative Performance Evaluation of POX and RYU Controllers 6. Analysis of Open vSwitch (OVS) Flow Tables and Switch Behavior in Mininet 7. Execution of REST API Operations on an SDN Controller (RYU/OpenDaylight) 8. Design and Deployment of a Custom Topology Script in Mininet	**30**

	9. Simulation of Traffic Monitoring in SDN Using RYU Simple Monitor Application 10. Mini-project and implementation and demonstration	
Pedagogy:	Instructional and Collaborative Learning, Inquiry Based Learning, Experimental Learning.	
References/ Readings:	**Textbooks:** 1. Paul Goransson, Chuck Black, Timothy Culver, Software Defined Networks: A Comprehensive Approach (2nd Edition), Morgan Kaufmann,2016, ISBN: 978-0128045558 2. Rajendra Chayapathy, et al, Network Function Virtualization (NFV) with a Touch of SDN, CRC Press,2020, ISBN: 978-0367352785 3. Thomas D. Nadeau, Ken Gray, SDN: Software Defined Networks – An Authoritative Review of Network Programmability Technologies, O'Reilly Media, 2013, ISBN: 978-1449342302 **Reference Books:** 1. Gerard Blanco, Radia Perlman, Interconnections of SDN and NFV, Addison-Wesley,2021, ISBN: 978-0139754851 2. KarthikRamasamy, S. Mathew,OpenFlow Cookbook, Packt Publishing, 2016, ISBN: 978-1783983744. 3. SrilathaVemula, SaiMugunth , Software Defined Networking with ONOS, Packt Publishing, 2017, ISBN: 978-1786465988. **Supplementary Web references** 1. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf 2. http://mininet.org/ 3. https://www.opennetworking.org/	
Course Outcomes:	**Upon completion of the course, students will be able to:** CO1**:** Analyze basic SDN topologies using Mininet and understand fundamental SDN concepts. CO2: Configure and test SDN controllers (POX/RYU) and demonstrate the implementation of OpenFlow rules and virtual network segmentation. CO3: Evaluate custom SDN network scenarios, including controller behavior, switch flows, REST APIs, and traffic monitoring. CO4: Apply SDN programmability to implement basic traffic management and redirection using Python-based controller scripts.	

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-331**
Title of the Course : **GPU Architecture and Programming**
Number of Credits : **3 (3L)**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of computer architecture, familiarity with operating systems.	
Course Objectives:	**This course will enable students to:** 1. Understand the basics of GPU architectures 2. Analyze the performance of GPU memory hierarchy. 3. Develop parallel programs using CUDA library. 4. Develop parallel programs using OpenCL library.	
Contents:		**No of Hours**
Unit-1	**Introduction of GPU architectures** – GPUs As Parallel Computers, Architecture of a Modern GPU, Why more speed or parallelism, Parallel programming Languages and Models. **History of GPU Computing** - Evolution of Graphical Pipelines, GPU computing, Future Trends.	**11**
Unit - 2	**Introduction to CUDA:** Data Parallelism, CUDA program Structure, Matrix- Matrix Multiplication, Device Memories and Data Transfer, Kernal Functions and Threading **CUDA Thread:** CUDA thread organization, Using BlockIdx and ThreadIdx, synchronization and Transparent Scalability, Thread Assignment, Thread Scheduling and latency Tolerance.	**11**
Unit - 3	**CUDA Memory:** Importance Access Efficiency, CUDA Device Memory types, A strategy for Reducing Global Memory Traffic, Memory as a Limiting factor to Parallelism **Floating Point Considerations:** Floating Point Format, Representable Numbers, Special Bit Patterns and Precision, Arithmetic Accuracy and Rounding.	**11**
Unit - 4	**Parallel Programming and computational Thinking:** Goals of parallel Programming, problem decomposition, algorithm selection, computational thinking. **Introduction to OpenCL:** Data parallelism Modem, Device Architecture, Kernel Function, Device Management and Kernel Launch,Electrostatic Potential Map in OpenCL	**12**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	

References/ Readings:	**Text Books** 1. David B. Kirk, Wen-meiW.Hwu: Programming Massively Parallel Processors: A Hands-On Approach ,Elsevier , 4th Edition ,2022, ISBN: 978-0-12-381472-2. 2. Shane Cook: CUDA Programming: —A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing), Elsevier, 2013, ISBN: 978-0-12-415933-4. **Reference Books** 1. David Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang, Heterogeneous Computing with OpenCL 2.0, Morgan Kaufmann, 2015, 3rd Ed, ISBN: 978- 0128014141 2. Jason Sanders, Edward Kandrot, —CUDA by Example: An Introduction to General Purpose GPU Programming‖, Addison - Wesley, 2010. ISBN:978-0131387683
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Describe GPU Architecture CO 2. Demonstrate programs using CUDA, identify issues and debug them. CO 3. Analyze the performance of GPU memory hierarchy. CO 4. Develop parallel programs using OpenCL library.

Name of the Programme : **B.E. in Computer Engineering**
Course Code : **CMP-332**
Title of the Course : **GPU Architecture and Programming Lab**
Number of Credits : **1P**
Effective from AY : **2024-25**

Pre-requisites for the Course:	Basic knowledge of computer architecture, familiarity with operating systems.	
Course Objectives:	**This course will enable students to:** 1. Introduce the fundamentals of GPU architecture, parallel execution models, and memory hierarchy used in modern many-core processors. 2. Develop the ability to design, implement, and optimize parallel programs using CUDA and OpenCL frameworks. 3. Enable students to analyze performance factors such as memory access patterns, thread organization, and workload distribution on GPU platforms. 4. Equip students with practical skills for solving compute-intensive problems through massively parallel processing and heterogeneous computing techniques.	
Content	**List of Experiments**	**No of Hours**
	1. Implementation of GPU Device Query Using CUDA 2. Vector Addition Using CUDA Kernels 3. Matrix Multiplication on GPU (Naïve and Shared-Memory Version) 4. Parallel Reduction (Sum) Using CUDA Shared Memory 5. Overlapping Data Transfers and Computation Using CUDA Streams 6. Platform and Device Identification Using OpenCL 7. Vector Addition Program Using OpenCL on CPU/GPU 8. Image Convolution (3×3 Filter) Using OpenCL 9. Effect of Work-Group Size on Kernel Performance in OpenCL 10. Performance Comparison of CPU vs GPU Execution Using OpenCL	**30**
Pedagogy:	Interactive, reflective, and inquiry-based methods, with a strong emphasis on critical thinking and problem-solving skills.	
References/ Readings:	**Text Books** 1. David B. Kirk, Wen-meiW.Hwu: Programming Massively Parallel Processors: A Hands-On Approach ,Elsevier , 4th Edition ,2022, ISBN: 978-0-12-381472-2.	

	2. Shane Cook: CUDA Programming: —A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing), Elsevier, 2013, ISBN: 978-0-12-415933-4. **Reference Books** 1. David Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang, Heterogeneous Computing with OpenCL 2.0, Morgan Kaufmann, 2015, 3rd Ed, ISBN: 978- 0128014141 2. Jason Sanders, Edward Kandrot, —CUDA by Example: An Introduction to General Purpose GPU Programming‖, Addison - Wesley, 2010. ISBN:978-0131387683
Course Outcomes:	Upon completion of the course, students will be able to: CO 1. Explain the architecture, execution model, and memory hierarchy of modern GPUs used in parallel processing. CO 2. Develop parallel programs using CUDA and OpenCL frameworks to solve compute- intensive problems. CO 3. Analyze the effect of thread organization, work-group sizing, and memory access patterns on GPU performance. CO 4. Evaluate CPU and GPU implementations to determine performance gains and optimization opportunities.