

USHA MARTIN UNIVERSITY, RANCHI
CENTRE FOR DISTANCE AND ONLINE
EDUCATION



PROGRAMME PROJECT REPORT
BACHELOR OF COMPUTER APPLICATIONS
(BCA)
2026–27

Registrar
कुल सचिव
Usha Martin University
उषा मार्टिन विश्वविद्यालय
Ranchi-835103 / राँची-835103

Introduction

The Bachelor of Computer Applications (BCA) Online Learning Programme is an innovative and flexible 4-year undergraduate (Honors) degree designed to equip students with a strong foundation in computer applications and software development. Tailored for modern learners, this programme leverages the advantages of online education to deliver a comprehensive curriculum that blends academic rigor with practical learning. Recognized for its academic depth, the BCA Online Learning Programme offers students a well-structured path toward advanced education and preparation for competitive examinations. By focusing on the fundamentals of computer science and applications, the programme provides learners with the essential knowledge and skills needed to excel in the evolving field of technology.

Delivered through an interactive online platform, the BCA Online Learning Programme incorporates cutting-edge ICT tools, enabling students to earn a minimum of 160 credits over four years. This flexible format empowers students to balance their studies with personal and professional commitments while ensuring access to high-quality education. The adaptability of the programme and focus on real-world applications make it highly relevant in today's digital age. Graduates of the BCA Online Learning Programme can explore diverse career opportunities, ranging from roles in the technology sector to academic pursuits. Additionally, the programme prepares students for competitive examinations, opening doors to rewarding positions in the government and public sectors.

With its emphasis on accessibility, academic excellence, and career readiness, the BCA Online Learning Programme is a gateway to a future rich with opportunities in technology, academia, and beyond.

Programme's Mission and Objectives

Mission:

The BCA Programme is dedicated to:

- Provide a high-quality, flexible, and accessible online learning experience in computer applications and software development, enabling students from diverse backgrounds to pursue their academic and career goals.
- Equip students with foundational and advanced knowledge in computer science, empowering them with the technical skills required to succeed in the dynamic IT industry.
- Instill culture of continuous self-improvement, ensuring that learners stay adaptable to the rapid advancements in technology and industry trends.
- Develop globally aware professionals who can navigate the challenges of an interconnected world and excel in international and multicultural settings.
- Bridge the gap between education and employability by providing students with the tools and skills needed for meaningful careers in the private and public sectors.

- Nurture a sense of integrity, ethical responsibility, and sustainability in students, ensuring their positive impact on society and the environment.
- Foster spirit of innovation and research among students, preparing them to contribute creatively to technological and societal advancements.

Objectives:

The objectives of the BCA Online Learning (OL) Programme are:

- Provide learners with a deep understanding of core concepts in computer applications, software development, and IT systems through a structured and flexible online curriculum.
- Equip learners with practical skills and proficiency in modern programming languages, database management, and emerging technologies, ensuring their preparedness for the IT industry.
- Offer an adaptable and technology-enabled learning platform that accommodates the diverse needs of students, including working professionals and those in remote locations.
- Build a strong academic foundation that supports students in pursuing higher education opportunities, such as MCA or MBA, and competitive exams for government and private sector roles.
- Instill a sense of responsibility, ethical conduct, and sustainability, ensuring that graduates contribute positively to their professions and communities.

A. Relevance of the Programme with Higher Education Institution’s Vision and Goals

The vision and mission of HEI, Usha Martin University of Ranchi are:

Vision:

To provide value-based education relevant for all, nurture local talent, support creation of excellence in teaching, learning and research, produce high quality innovative graduates and contribute towards sustainable development of the state.

Mission:

The University shall strive to promote innovative strategies for seamless dissemination and creation of knowledge using latest techniques, available media and technologies so that its graduates acquire skills to get sustainable employment and contribute with a sense of service to national and global society.

Goals:

The relevance of the BCA programme to Higher Education Institutions' (HEIs) goals lies in its alignment with the broader mission and objectives of the institution. Here's how the programme can contribute to the HEI's goals:

- **Academic Excellence:** The BCA Online Learning Programme enhances the institution's reputation by providing a high-quality, technology-enabled education in computer science and IT. It attracts talented learners from diverse backgrounds and produces graduates equipped to excel in their careers and adapt to evolving industry demands.

- **Research and Innovation:** By fostering a culture of curiosity and creativity in an online setting, the programme encourages students and faculty to engage in innovative research and collaborate with industry partners to develop practical solutions to real-world challenges.
- **Community Engagement:** Through virtual service-learning initiatives, collaborative projects, and online outreach programs, the BCA Online Learning Programme strengthens the institution's ties to the community, promoting social responsibility and civic engagement in a digital context.
- **Global Perspective:** The programme integrates global perspectives by offering multicultural virtual learning environments and opportunities for international collaborations, preparing students to thrive in an increasingly interconnected and digitalized world.
- **Employability and Career Development:** Designed to enhance students' employability, the programme equips learners with industry-relevant skills and knowledge through online courses, virtual internships, and project-based learning, ensuring they are ready to succeed in a competitive job market.
- **Diversity and Inclusion:** By leveraging the flexibility of online learning, the programme promotes equity and inclusivity, making quality education accessible to students from diverse geographic, cultural, and socioeconomic backgrounds.
- **Continuous Improvement:** The programme employs ongoing assessment, evaluation, and feedback mechanisms to ensure it remains responsive to the changing needs of students, industry trends, and technological advancements, fostering continuous growth and excellence.

Overall, the BCA programme can serve as a key driver of the institution's goals, enhancing its reputation, impact, and contribution to the broader community.

C. Nature of Prospective Target Group of Learners

The Online Learning (OL) programmes of Usha Martin University (UMU) shall target the working professionals, executives as well as those who cannot attend a full-time programme due to prior occupation or other assignments. The candidates desirous of taking admission in Bachelor in Computer Applications (BCA) programme shall have to meet the eligibility norms as follows –

- To obtain admission in BCA programme offered through OL mode, the learner must have completed (10+2) or equivalent in science stream or equivalent course.
- The OL BCA programme offered by Usha Martin University caters the needs of diverse groups of undergraduate learners from all disciplines located in diverse regions and social structures such as learners from a low level of disposable income, rural dwellers, women and minorities who have little access to formal institutions of higher learning.

D. Appropriateness of Programme to be conducted in OL mode to acquire specific skills and competence

The University has identified the following Programme Outcomes (POs) and Programme Specific Outcomes (PSOs) as acquisition of specific skills and competence in BCA Programme.

Programme Outcomes (POs)

After completing the BCA programme, students will be able to:

PO1. Computational knowledge: Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.
PO2. Design/development of solutions: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO3. Modern tool usage: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PO4. Societal and Environmental Concern: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.
PO5. Professional Ethics: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.
PO6. Individual and team work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
PO7. Communication efficacy: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO8. Project management and finance: Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO9. Life-long learning: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

Programme Specific Outcomes (PSOs)

PSO1. Ability to understand and apply knowledge on analysis, design and development of software applications.
PSO2. Ability to work with latest computing technologies and pursue careers in IT industry, teaching and allied areas.

E. Instructional Design

The BCA programme is structured into four semesters, with a minimum credit requirement of 160 to obtain the degree. In OL (Online Learning) mode from Usha Martin University, the minimum time

period for completing the BCA degree is four years, while the maximum allowable time period is eight years.

1. Programme Structure and Credit Framework

- The programme is structured in a semester-based modular format with defined credits aligned with UGC norms and NEP 2020 guidelines.
- Each course is divided into:
 - Blocks → Units → Topics → Learning Activities
- Credit distribution ensures a balance between:
 - Theory
 - Practical/Lab work
 - Projects and internships
- Provision for:
 - Multiple entry-exit options (if applicable)
 - Credit transfer and accumulation (ABC framework)

2. Learning Design Approach (Pedagogical Framework)

The instructional design follows a learner-centric and outcome-based approach, integrating:

- **Outcome-Based Education (OBE)**
 - Alignment of Course Outcomes (COs), Programme Outcomes (POs), and PSOs
- **Constructivist Learning Theory**
 - Encouraging active participation and knowledge construction

Combination of synchronous and asynchronous learning

3. Mode of Content Delivery

The programme adopts a multi-mode digital delivery system:

a) E-Content (Quadrant I: Digital Self-Learning Material)

- Developed as per UGC-DEB guidelines
- Features:
 - Simple and interactive language
 - Learning objectives at the beginning
 - In-text questions and summaries
 - Case studies or examples
 - Glossary and references

b) Interactive Video (Quadrant II: e-Tutorials / Video Lectures)

Video lectures enhance conceptual clarity and simulate classroom experience.

Types of Video Content:

- **Concept-based lectures** (theoretical explanations)
- **Demonstration videos** (coding, simulations, tools like Python, R)

Design Features:

- Duration: 5 Hours per 1 Credit Course which can be subsequently divided into blocks and units.
- Chunking of topics for better retention
- Subtitles and transcripts for accessibility
- Use of:
 - Screen recording (for programming)
 - Simulation Tools (for algorithms, networks)
 - Real-life problem-solving demonstrations

Interactive Elements:

- In-video quizzes (pause-and-answer)
- Scenario-based questions

c) Quizzes and Assignments (Quadrant III: Assessment & Evaluation)

Continuous assessment ensures learning reinforcement and outcome achievement.

Types of Assessments:**i. Quizzes:**

- MCQs, True/False, Match the Following
- Auto-graded through LMS
- Topic-wise and module-wise quizzes
- Timed assessments for self-evaluation

ii. Assignments:

- Descriptive/problem-solving questions
- Programming assignments (C, C++, Python, Java)
- Case study analysis (e.g., cybersecurity breach case)
- Mini-projects (e.g., web development, data analytics)

iii. Practical/Lab Work:

- Virtual lab assignments
- Simulation-based exercises
- Code submission and evaluation

iv. Assessment Design Features:

- Aligned with Bloom's Taxonomy (The nomenclature is defined as follows: L1 – Remember, L2 – Understand, L3 – Apply, L4 – Analyze, L5 – Evaluate, and L6 – Create.)
- Rubric-based evaluation
- Feedback mechanism (automated + faculty)

d) Discussion Forums (Quadrant IV: Interaction & Collaboration)

Discussion forums promote collaborative learning and academic engagement.

Types of Forums:

- Course-wise discussion boards
- Topic-specific threads (e.g., AI, DBMS, Networks)
- Doubt-clearing forums
- Peer discussion groups

Key Features:

- Moderated by faculty/subject experts
- Encourages:
 - Question-answer interactions
 - Idea sharing and problem-solving
- Weekly discussion prompts
- Integration with LMS notifications

Engagement Strategies:

- Case-based discussions
- Group discussions and debates
- Problem-solving threads (coding/debugging help)
- Live doubt-clearing sessions
- Chat-based instant support

4. Teaching-Learning Strategies

A variety of instructional methods are used:

- **Synchronous Learning**
 - Live online lectures
 - Webinars and expert sessions
 - Virtual classrooms
- **Asynchronous Learning**
 - Recorded lectures
 - Discussion forums
 - Self-paced modules
- **Experiential Learning**
 - Case studies
 - Simulations
 - Industry-based projects
- **Collaborative Learning**
 - Group discussions
 - Peer learning activities
 - Team-based assignments

5. Practical and Skill-Based Learning

- **Virtual Labs and Simulation Tools**
- Programming practice platforms
- Real-time coding environments

- Industry-relevant tools (e.g., IDEs, DBMS tools)
- **Project-Based Learning**
 - Mini projects (semester-wise)
 - Major project (final semester)
 - Problem-solving and innovation focus

6. Learner Support Services

To ensure effectiveness of online learning:

- **Academic Support**
 - Faculty mentoring
 - Doubt-clearing sessions
 - Discussion boards
- **Technical Support**
 - LMS helpdesk
 - IT support team
- **Administrative Support**
 - Admission and examination helpdesk
- **Counselling Services**
 - Career guidance

7. Induction and Orientation Programme

- Orientation on:
 - LMS usage
 - Online learning strategies
 - Programme structure
- Awareness about:
 - Academic integrity
 - Digital etiquette

8. Feedback and Continuous Improvement

- Regular learner feedback surveys
- Course improvement cycles
- Faculty development programmes (FDPs)

Course structure and Syllabi (2026-27) Evaluation Scheme

Three Year BCA Programme:

The total credits for 3-year BCA will be minimum 120. Following types of courses will be offered for a 3-Year BCA Programme.

- 15 Discipline-specific Major Courses (60 credits)
- 9 Interdisciplinary and Minor Courses (33 credits)
- 4 Ability Enhancement Courses (8 credits)
- 3 Skills Enhancement Courses (10 credits)
- 2 Value-added Courses (6 credits)
- 1 Internship (3 credits)

Four Year BCA (Hons.) Programme:

The 4-year BCA (Hons) degree will be minimum 160. Following types of courses will be offered for a 4-Year BCA(H) Programme:

- 20 Discipline-specific Major Courses (92 credits)
- 11 Interdisciplinary and Minor Courses (41 credits)
- 4 Ability Enhancement Courses (8 credits)
- 3 Skill Enhancement Courses (10 credits)
- 2 Value-added courses (6 credits)
- 1 Internship (3 credits)

Category wise Credit

Sl. No.	Course Name	Category	No of Courses	Credit	Total Credit	No of Courses	Credit	Total Credit
1	Discipline Specific Courses - Core Major (Core) Course	CC	15	60	120	20	92	160
2	Inter disciplinary Minor	IDC	9	33		11	41	
3	Ability Enhancement Course	AEC	4	8			8	
4	Skill Enhancement Course	SEC	3	10			10	
5	Common Value-added Courses	VAC	2	6			6	
6	Project and Internship		1	3			3	

Year and Credit distribution:

Bachelor of Computer Applications (BCA)								
Year	CC	IDC & Minor	AEC	SEC	VAC	Internship	Total Credit	Exit Option
1	16	16	4	4	0	0	40	40
2	16	8	4	6	6	0	40	80
3	28	9	0	0	0	3	40	120
Total	60	33	8	10	6	3	120	120
4	32	8	0	0	0	0	40	160
Total	92	41	8	10	6	3	160	160

Semester – I

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1101	Management Information System	Major	4	30	70	100
2.	CABO-1102	Problem solving Using 'C'	Major	2	30	70	100
3.	CABO-1103	Mathematics-I	Minor	4	30	70	100
4.	CABO-1104	Financial Accounting	Minor	4	30	70	100
5.	ENBO-1101/H NBO-1101	English Communication/ रचनाक और समाचार लेखन	AEC	2	30	70	100
6.	CABO-1105	Computer Fundamental & Office Automation	SEC	2	30	70	100
7.	CABO-1116	Problem solving Using 'C' Lab	Major	2	30	70	100
Total				20	210	490	700

Note: Students are advised to opt Minor courses as given in the evaluation scheme. However, student is free to opt any other minor courses (as offered by the other departments) as per his/her interest/choice.

Semester – II

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1201	System Analysis & Design	Major	4	30	70	100
2.	CABO-1202	C++ & Data Structure	Major	2	30	70	100
3.	CABO-1203	Ordinary Differential Equations	Minor	4	30	70	100
4.	CABO-1204	Fundamentals of Electronics	Minor	4	30	70	100
5.	ENBO-1201 or HNBO-1201	Creative Writing/ िफj और मीिड या लेखन	AEC	2	30	70	100
6.	CABO-1205	Logical Reasoning	SEC	2	30	70	100
7.	CABO-1216	C++ & Data Structure Lab	Major	2	30	70	100
Total				20	210	490	700

Note: The student taking an exit from programming after securing 40 credits of the first and second semester shall be awarded a Certificate in Computing (CIC).

Semester – III

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1301	Computer Organization	Major	4	30	70	100
2.	CABO-1302	Operating System	Major	2	30	70	100
3.	CABO-1303	Integral Calculus	Minor	4	30	70	100
4.	ENBO-1301/ HNBO-1301	Personality Development/ पटकथा लेखन	AEC	2	30	70	100
5.	VACO-1301	Environmental Education	VAC	3	30	70	100
6.	CABO-1304	Computer Based Numerical Methods	SEC	3	30	70	100
7.	CABO-1315	Operating System Lab	Major	2	30	70	100
Total				20	210	490	700

Semester – IV

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1401	Analysis and Design of Algorithms	Major	4	30	70	100
2.	CABO-1402	Object Oriented Programming with Java	Major	2	30	70	100
3	CABO-1403	Matrix Theory	Minor	4	30	70	100
4.	CABO-1404	Database Management System	SEC	3	30	70	100
5.	ENBO-1401	Basic Knowledge of English Grammar	AEC	2	30	70	100
6.	VACO-1401	Understanding India	VAC	3	30	70	100
7.	CABO-1415	Object Oriented Programming with Java Lab	Major	2	30	70	100
Total				20	210	490	700

Note: The students taking an exit from the programme after securing 80 credits from first to fourth semester shall be awarded a Diploma in Computer Applications (DCA).

Semester – V

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1501	Software Engineering	Major	4	30	70	100
2.	CABO-1502	Computer Networks	Major	4	30	70	100
3.	CABO-1503	Theory of Computation	Major	4	30	70	100
4.	CABO-1504	Perspective of Physics	Interdisciplinary	3	30	70	100
5.	CABO-1505	Basic Abstract & Linear Algebra	Interdisciplinary	3	30	70	100
6.	CABO-1516	Software Engineering Lab	Major	2	30	70	100
Total				20	180	420	600

Semester – VI

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1	CABO-1601	IoT & Cloud Computing	Major	4	30	70	100
2	CABO-1602	Machine Learning Using Python	Major	4	30	70	100
3	CABO-1603	Discrete Mathematics	Major	4	30	70	100
4	CABO-1604	Concepts of Physics	Interdisciplinary	3	30	70	100
5	CABO-1615	Machine Learning Using Python Lab	Major	2	30	70	100
6	CABO-1691	Internship	SEC	3	30	70	100
Total				20	180	420	600

Note: The students who want to undertake 3 years Under graduate Programme will be awarded Bachelor of Computer Applications (BCA) degree upon securing 120 credits.

Semester – VII

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1	CABO-1701	Soft Computing	Major	4	30	70	100
2	CABO-1702	R Programming	Major	4	30	70	100
3	CABO-1703	Network Security & Cryptography	Major	4	30	70	100
4.	CABO-1704	Parallel Computing	Major	4	30	70	100
5.	CABO-1705	Linear Programming	Minor	4	30	70	100
Total				20	150	350	500

Semester – VIII

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CABO-1801	Natural Language Processing	Major	4	30	70	100
2.	CABO-1802	Web Designing	Major	4	30	70	100
3.	CABO-1803	Data Warehousing & Data Mining	Major	4	30	70	100
4.	CABO-1804	Digital Forensics	Major	4	30	70	100
5.	CABO-1805	Bioinformatics using AI	Minor	4	30	70	100
Total				20	150	350	500

Note: Students will be awarded Bachelor in Computer Applications (Honours) Degree upon securing 160 credits.

SEMESTER – I

Course Code: CABO–1101

Credit: 4

Course Name: Management Information System

Course Objectives:

- To equip students with the knowledge and skills necessary to understand, design, and utilize information systems effectively in organizations.
- To provide insights into the role of MIS in facilitating managerial decision-making and enhancing organizational efficiency.
- To offer comprehensive understanding of key MIS concepts, including information technology infrastructure, database management, and system analysis and design.
- To explore the strategic role of information systems in gaining competitive advantage and improving organizational management.
- To combine theoretical lectures, practical cash flow studies, and hands-on projects to ensure a well-rounded learning experience in MIS.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts, principles, and terminology related to the subject.	L2
CO2	Apply appropriate methods, algorithms, or techniques to solve structured computing problems.	L3
CO3	Analyze problems to identify requirements, constraints, and suitable computational approaches.	L4
CO4	Evaluate different solutions, tools, or methodologies based on performance, efficiency, and feasibility.	L5
CO5	Design and develop a system/module/project to solve real-world problems using appropriate technologies.	L6

Block I: Introduction to Management Information Systems

Unit 1: Overview of Management Information Systems: Definition and scope of MIS, Importance of MIS in organizations, Historical development and evolution of MIS

Unit 2: Information Technology Infrastructure: Components of information technology infrastructure, Hardware, software, networks, and telecommunications, Cloud computing and emerging trends in IT infrastructure

Unit 3: Database Management Systems (DBMS): Introduction to database concepts, Relational database management systems (RDBMS), Database design, normalization, and query languages

Unit 4: Systems Development Life Cycle (SDLC): Overview of SDLC phases: planning, analysis, design, implementation, and maintenance, Approaches to system development: waterfall, agile, and iterative.

Block II: System Analysis and Design

Unit 5: Requirements Analysis and Modeling: Elicitation and documentation of user requirements, Use CABOe diagrams, entity-relationship diagrams (ERD), and data flow diagrams (DFD)

Unit 6: System Design and Architecture: Design principles and methodologies, Architectural models: client-server, peer-to-peer, and cloud-based architectures

Unit 7: Information Systems Security and Risk Management: Threats to information security, Risk assessment and mitigation strategies, Security policies, procedures, and controls

Unit 8: IT Governance and Compliance: Principles of IT governance, Regulatory compliance (e.g., GDPR, HIPAA), IT audit and assurance

Block III: Managing Information Systems

Unit 9: Project Management in Information Systems: Project planning, scheduling, and resource allocation, Project management methodologies (e.g., PMBOK, PRINCE2), Risk management and project success factors

Unit 10: Strategic Role of Information Systems: Aligning IT with business strategy, Competitive advantage through IT-enabled innovation, Digital transformation and disruptive technologies

Unit 11: Business Intelligence and Analytics: Data-driven decision-making, Data warehousing and data mining techniques, Business analytics tools and techniques

Unit 12: Enterprise Systems and ERP: Overview of enterprise systems, Enterprise Resource Planning (ERP) systems, Implementation challenges and best practices

Block IV: Strategic Management of Information Systems

Unit 13: Emerging Trends in MIS: Internet of Things (IoT) and connected devices, Artificial Intelligence (AI) and machine learning in MIS, Blockchain technology and decentralized applications

Unit 14: E-Business and E-Commerce: Fundamentals of e-business and e-commerce, E-commerce platforms and business models, Legal and ethical issues in e-commerce

Unit 15: Knowledge Management Systems: Introduction to knowledge management, Knowledge management tools and techniques, Implementing and evaluating knowledge management systems

Unit 16: CABOe Studies and Future Directions in MIS: CABOe studies of MIS in various industries, Future directions and trends in MIS, Integrative project on MIS strategy

References:

1. K. C. Laudon and J. P. Laudon, "Management Information Systems: Managing the Digital Firm", 16th Edition, Pearson, 2020.
2. G. B. Davis and M. H. Olson, "Management Information Systems: Conceptual Foundations, Structure and Development", 2nd Edition, McGraw Hill, 1985.
3. R. Murdick, J. Ross and J. Claggett, "Information Systems for Modern Management", 3rd Edition, PHI, 2000.
4. S. Sadagopan, "Management Information Systems", 3rd Edition, PHI Learning, 2014.
5. J. O'Brien and G. Marakas, "Management Information Systems", 10th Edition, McGraw Hill, 2011.

Course Code: CABO–1102

Credit: 2

Course Name: Problem solving Using ‘C’

Course Objectives:

- To introduce students to programming principles and foundational concepts essential for software development.
- To teach programming fundamentals, including variables, control structures, functions, and data types.
- To explore algorithm design and analysis techniques, focusing on problem-solving strategies and algorithmic efficiency.
- To develop skills through practical coding exercises and algorithmic challenges.
- To equip students to write efficient, structured code and solve computational problems effectively.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental programming concepts including algorithms, data types, control structures, arrays, strings, and basic data structures.	L2
CO2	Apply C programming constructs such as functions, arrays, pointers, structures, and file handling to solve computational problems.	L3
CO3	Analyze problems using appropriate data structures and algorithmic techniques including recursion, searching, and sorting methods.	L4
CO4	Evaluate the efficiency of algorithms and programs based on time and space complexity, debugging techniques, and optimization methods.	L5
CO5	Design and develop efficient C programs implementing data structures and algorithms to address real-world computing problems.	L6

Block I: Foundations of Programming

Unit 1: Introduction to Programming: Overview of programming paradigms, Basics of algorithmic problem-solving, Introduction to programming languages and development environments

Unit 2: Fundamentals of Programming: Variables, data types, and operators, Control structures: selection and iteration, Functions and modular programming concepts

Unit 3: Arrays and Strings: Introduction to arrays and strings, Array manipulation and string handling operations, Multi-dimensional arrays and array algorithms

Unit 4: Introduction to Algorithms: Basic algorithm analysis, Introduction to algorithm design techniques, Recursion and recursive algorithms

Block II: Arrays and Strings

Unit 5: Arrays in C, Introduction to arrays: declaration, initialization, accessing elements, Array manipulation: sorting, searching, merging, Multi-dimensional arrays and array of strings

Unit 6: Character Strings Handling, Introduction to character strings: declaration, initialization, String input-output functions: gets, puts, String manipulation functions: strlen, strcpy, strcat, etc.

Unit 7: Pointers and Dynamic Memory Allocation, Understanding pointers: declaration, initialization, arithmetic operations, Dynamic memory allocation functions: malloc, calloc, realloc, free

Applications of pointers and dynamic memory allocation in solving problems

Unit 8: Structures and File Handling, Introduction to structures: declaration, definition, accessing members, Array of structures and structure within structures, File handling in C: opening, reading, writing, and closing files

Block III: Advanced C Programming Concepts

Unit 9: Advanced Control Structures, Nested loops and nested decision-making statements

Loop control statements: break, continue, Recursion: principles, advantages, limitations

Unit 10: Advanced Functions, Function pointers and callback functions, Variables functions and their applications, Understanding scope and lifetime of variables

Unit 11: Preprocessor Directives and Macros, Overview of preprocessor directives: #include, #define, #ifdef, #ifndef, Macros and their applications in code optimization and customization, Conditional compilation and header file management

Unit 12: Error Handling and Debugging Techniques, Error handling techniques: return values, errno, assert, debugging tools and techniques: printf debugging, gdb debugger, Strategies for identifying and resolving common programming errors

Block IV: Data Structures and Basic Algorithms in C

Unit 13: Introduction to Data Structures, Overview of data structures: arrays, linked lists, stacks, queues

Understanding the importance of data structures in problem-solving

Unit 14: Implementing Data Structures in C, Implementation of linked lists, stacks, and queues in C

Operations on data structures: insertion, deletion, traversal, Analyzing the time and space complexity of operations on data structures

Unit 15: Searching and Sorting Algorithms, Implementation and analysis of searching algorithms: linear search, binary search, Implementation and analysis of sorting algorithms: bubble sort, insertion sort, selection sort, merge sort, quick sort

Unit 16: Advanced Data Structures and Algorithm Analysis, Introduction to advanced data structures: trees, graphs, hash tables, Analyzing the efficiency of algorithms: time complexity, space complexity

Practical applications of data structures and algorithms in solving real-world problems

References:

1. E. Balagurusamy, “Programming in ANSI C”, 8th Edition, McGraw Hill, 2019.
2. Byron Gottfried, “Programming with C”, 3rd Edition, McGraw Hill, 2018.
3. Reema Thareja, “Programming in C”, 2nd Edition, Oxford University Press, 2021.
4. Yashavant Kanetkar, “Let Us C”, 17th Edition, BPB Publications, 2023.
5. J. R. Hanly and E. B. Koffman, “Problem Solving and Program Design in C”, 8th Edition, Pearson, 2022.

Course Code: CABO-1103

Credit: 4

Course Name: Mathematics-I

Course Objectives:

- To provide students with a foundational understanding of essential mathematical concepts.
- To ensure comprehension of sets and functions, properties of numbers, matrices, and systems of linear equations.
- To teach key calculus concepts, including limits, continuity, differentiability, and integration.
- To prepare students for further studies in mathematics and related fields.
- To enable students to analyse problems and apply mathematical techniques with confidence.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of sets, relations, functions, mathematical induction, and number theory including GCD and Diophantine equations.	L2
CO2	Apply matrix operations, determinants, and methods to solve systems of linear equations.	L3
CO3	Analyze functions to determine limits, continuity, and differentiability, and apply rules of differentiation including higher-order derivatives.	L4
CO4	Evaluate mathematical problems using appropriate techniques of differentiation and integration, including substitution and integration by parts.	L5
CO5	Solve complex real-world and theoretical problems by integrating concepts of algebra, matrices, and calculus.	L6

Block I: Introduction to sets functions and numbers

Unit 1. Sets, relation and function

Unit 2. Principle of Mathematics Induction

Unit 3. Division Algorithm, Greatest Common Divisor, The Euclidean Algorithm

Unit 4. Linear Diophantine Equations

Block II: Introduction to Matrix Theory

Unit 5. Matrices: Types and Basic Properties

Unit 6. Matrix Multiplication and Determinant

Unit 7. Rank of the Matrix

Unit 8. System of Linear Equations

Block III: Limit, Continuity and Differentiability of Function

Unit 9. Limit of a Function and Algebra of Limits

Unit 10. Continuous Functions

Unit 11. Differentiable Functions

Unit 12. Successive Differentiation and Leibnitz Theorem

Block IV: Integration

Unit 13. Integration as Inverse of Differentiation

Unit 14. Method of Integration by Substitution

Unit 15. Method of Integration by Parts

Unit 16. Some Special Integrals

References:

1. G. B. Thomas and R. L. Finney, "Calculus and Analytic Geometry", 14th Edition, Pearson, 2021.
2. Erwin Kreyszig, "Advanced Engineering Mathematics", 10th Edition, Wiley India, 2020.
3. B. S. Grewal, "Higher Engineering Mathematics", 44th Edition, Khanna Publishers, 2023.
4. Shanti Narayan and P. K. Mittal, "Matrices", 12th Edition, S. Chand & Company, 2022.
5. S. Lipschutz and M. Lipson, "Linear Algebra (Schaum's Outline Series)", 6th Edition, McGraw Hill, 2021.

Course Code: CABO-1104

Credit: 4

Course Name: Financial Accounting

Course Objectives:

- To provide students with a comprehensive understanding of the principles and concepts of financial accounting.
- To teach the methods of recording, classifying, and summarizing financial transactions.
- To develop the ability to prepare and interpret financial statements, including balance sheets, income statements, and CABO_h flow statements.
- To enhance students' skills in analyzing financial data to make informed business decisions.
- To ensure students understand the regulatory environment and ethical considerations in financial accounting.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental accounting concepts, principles, conventions, accounting standards, and the accounting equation.	L2
CO2	Apply accounting procedures to record business transactions using journal, ledger, cash book, and prepare trial balance.	L3
CO3	Analyze financial data to prepare and interpret financial statements of sole proprietorship firms with adjustments.	L4
CO4	Evaluate accounting treatments related to depreciation, hire purchase system, and plant property and equipment as per relevant standards.	L5
CO5	Prepare and solve complex accounting problems involving branch accounting, departmental accounting, and royalty accounts.	L6

Block I: Introduction

Unit 1: Conceptual Framework: Book keeping, Accounting & Accountancy, objectives, functions, advantage, limitations,

Unit 2: Accounting principle, Concepts and Conventions, Accounting Equations,

Unit 3: Introduction to Accounting Standards and Indian Accounting Standards (AS & Ind. AS).

Block II: Accounting Process

Unit 4: Journal, ledger, CABO_h Book, Trial Balance,

Unit 5: Preparation of Financial Statements of a profit making sole proprietorship trading firm with additional information

Unit 6: Preparations of Final Accounts.

Block III: Depreciation and Hire Purchase Accounting

Unit 7: Accounting for Plant Property and Equipment

Unit 8: Depreciation: Meaning of Depreciation, Objective and Methods of depreciation (Straight line, Diminishing Balance), Change of Method. (Relevant accounting Standards as applicable)

Unit 9: Hire Purchase Accounting: Calculation of Interest, Partial and Full Repossession, profit Computation (Stock & Debtors System only), Accounting for Installment System (Simple practical problems)

Block IV: Special Types of Accounting

Unit 10: Accounting for Branches (excluding foreign branches): Dependent branches ('Debtors system' and 'Stock & debtors System') and overview of Independent branches.

Unit 11: Departmental Accounting: Concept, Type of departments, Basis of allocation of departmental expenses

Unit 12: Methods of departmental accounting (Relevant accounting Standards as applicable)

Block V: Royalties Accounts

Unit 13: Royalty account, Minimum Rent,

Unit 14: Computation and recovery of Short working in the books of Land lord etc.

Unit 15: Practical questions

References:

1. T. S. Grewal, "Double Entry Book Keeping – Financial Accounting", 2023 Edition, Sultan Chand & Sons, New Delhi.
2. S. N. Maheshwari and S. K. Maheshwari, "Financial Accounting", 7th Edition, Vikas Publishing House, 2021.
3. P. C. Tulsian and Bharat Tulsian, "Financial Accounting", 3rd Edition, Pearson Education, 2022.
4. R. Narayanaswamy, "Financial Accounting: A Managerial Perspective", 7th Edition, PHI Learning, 2023.
5. Ashok Sehgal and Deepak Sehgal, "Fundamentals of Financial Accounting", 2nd Edition, Taxmann Publications, 2021.

Course Code: ENBO–1101

Credit: 2

Course Name: English Communication

Course Objectives:

- To equip students with essential skills and confidence for effective English communication.
- To develop proficiency in speaking, listening, reading, and writing.
- To enhance interpersonal communication abilities in various contexts.
- To improve professional communication skills.
- To provide a comprehensive curriculum to support language development.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamentals of communication, including flow, types, and elements of non-verbal communication.	L2
CO2	Apply non-verbal communication skills such as body language and paralanguage in interpersonal interactions.	L3
CO3	Analyze communication patterns and behaviors in real-life situations such as meetings, travel, and social interactions.	L4
CO4	Evaluate the effectiveness of communication strategies used in group discussions and everyday situations.	L5
CO5	Demonstrate effective communication skills in real-life scenarios including meeting people, shopping, travelling, and group discussions.	L6

Block I. Non-Verbal Communication

Unit 1 Definition of Communication

Unit 2 Flow of communication

Unit 3 Types of non- verbal communication

Unit 4 Body Language

Unit 5 Paralanguage skills

Block II. Conversation in Real Life Situations

Unit 6 Meeting people

Unit 7 Travelling

Unit 8 Visiting Places

Unit 9 Shopping

Unit 10 Group Discussion

References:

1. Guffey, M. E., & Loewy, D., "Business Communication: Process and Product", Cengage Learning, 2019.
2. Lesikar, R. V., Flatley, M. E., & Rentz, K., "Basic Business Communication: Skills for Empowering the Internet Generation", McGraw-Hill Education, 2017.
3. O'Hair, D., Wiemann, M., & Mullin, D. I., "Real Communication: An Introduction", Bedford/St. Martin's, 2018.
4. Adler, R. B., & Elmhorst, J. M., "Communicating at Work: Principles and Practices for Business and the Professions", McGraw-Hill Education, 2016.
5. Hargie, O., "Skilled Interpersonal Communication: Research, Theory and Practice", Routledge, 2019.

उद्देश्य

हिंदी की मुख्य विधाओं की लेखन-प्रक्रिया से परिचित करवाते 5 लेखन-

अपेक्षाओं की लेखन-

प्रतिभा को निखारना एवं उच्च सृजनक लेखन हेतु प्रेरित करना। समाचार लेखन से परिचित कराना।

अधिगम की उपलियां

CO1	रचनाक लेखन एवं समाचार लेखन की मूल अवधारणाओं, 4Vप, उंकार एवं तर्कों को समझना।	L2
CO2	किवता, लघुकथा, यात्रा-वृत्त एवं रिपोर्टेज लेखन की विधियों का आवहारक अनुपयोग करना।	L3
CO3	विभिन्न उंकार के समाचारों, पत्रों तथा संवाददाता की भूमिका का विश्लेषण करना।	L4
CO4	समाचार लेखन एवं रचनाक लेखन की गुणवत्ता एवं उभावशीलता का मूलांकन करना।	L5
CO5	रचनाक एवं प्रकाशिता लेखन के विभिन्न रूपों (किवता, लघुकथा, समाचार आदि) का सृजन करना।	L6

खंड-1 रचनाक लेखन

इकाई-1 रचनाक लेखन : अथर्व 4Vप

इकाई-2 किवता लेखन: 4Vप एवं अपेक्षा

इकाई-3 लघुकथालेखन: 4Vप एवं अपेक्षा

इकाई-4 यात्रावृत्त

लेखन: 4Vप एवं अपेक्षा इकाई-5 रिपोर्टेज-

लेखन: 4Vप एवं अपेक्षा

खंड-2 समाचार लेखन

इकाई-6समाचार:अथN,पृरभाषाएवंतe

इकाई-7समाचारकेपुंकार, ढोत

इकाई-8संवाददाता:गुणऔरपुंकार

इकाई-9समाचारलेखन-अढास१

इकाई-10समाचारलेखन-अढासर

सहायकपु क

1. डॉ. नगेर, “सृजनाढक लेखन”, राजकमल पुंकाशन, 2018।
2. डॉ. िवन्नाथ िठपाठी, “आलोचना और रचनाढकता”, वाणी पुंकाशन, 2020।
3. डॉ. रामजी ितवारी, “पढकाऱता िस5ांत और ढवहार”, राजपाल ँड संज़, 2019।
4. पुढेश पंत, “समाचार लेखन की कला”, नेशनल पिढिशिंग हाउस, 2021।
5. डॉ. सिढड दानंद पाढेय, “िहंदी पढकाऱता: इतहास, 4Vप और लेखन कला”, भारती भवन, 2022।

Course Code: CABO–1105

Credit: 2

Course Name: Computer Fundamental & Office Automation

Course Objectives:

- To introduce students to basic computer concepts and operating systems.
- To teach the fundamental principles of hardware and software components.
- To develop proficiency in using office automation tools such as word processors, spreadsheets, and presentation software.
- To enhance understanding of computer networks and internet applications.
- To equip students with practical skills for efficiently managing and automating office tasks.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of computers, including history, characteristics, data processing, and types of storage devices.	L2
CO2	Apply knowledge of computer hardware, software, operating systems, and input-output devices in basic computing tasks.	L3
CO3	Analyze data organization, file systems, and different data processing methods in computer systems.	L4
CO4	Evaluate the performance and suitability of storage devices, operating systems, and input-output devices for different applications.	L5
CO5	Create professional documents, spreadsheets, and presentations using MS Word and MS Excel tools for real-world applications.	L6

Block I: INTRODUCTION TO COMPUTER & STORAGE DEVICES

Unit I: Brief history of development of computers, computer system concept, characteristics, capabilities and limitations, types of computers.

Unit II: BIOS, Software, Hardware, Firmware, Booting files & Directory system. Data, information and their need, Levels of information, Quality of information, Comparison of manual & electronic storage of data,

Unit III: Organization of data as file, Use of information in data processing systems, various data processing methods.

Unit IV: Primary Storage: Storage locations and addresses, storage capacity, RAM, ROM, PROM, EPROM, Cache memory.

Unit V: Secondary Storage: Sequential & Direct Access devices, Punched paper devices Magnetic tape, Magnetic Disk, Floppy Disk, Optical Disk, Magnetic Bubble Memory.

Block II: INPUT-OUTPUT, OPERATING SYSTEM & OFFICE

Unit VI: Input-Output devices: Keyboard, Pointing Devices: Mouse Trackball, Touch pad, Track point, Joystick, Touch Screen, Scanner, Barcode Reader, Optical Mark Reader.

Unit VII: Basic Elements, Functions and Types of Operating System, Serial Processing, Multi-Programmed, Batch System.

Unit VIII: Time Sharing Systems, System Components, Operating System Services, Interrupts, Interrupt Processing,

Unit IX: MS-Office, Introduction to MS-Word menus shortcuts, create a word document, opening a file-saving, editing text documents, cut, copy, paste, formatting a document, alignments, font styles, indents. Creating tables – merging, splitting, drawing-shapes, picture tools, mail merge, spell check.

Unit X: MS-Excel, Introduction, working spread sheets, formatting spread sheets, creating charts, formula usage.

References:

1. Norton, P., “Introduction to Computers: Office Automation”, Pearson, 2018.
2. Shelly, G., Cashman, T. J., & Vermaat, M. E., “Discovering Computers & Microsoft Office 365 & Office 2016: A Fundamental Combined Approach”, Cengage Learning, 2016.
3. Sinha, P. K., “Computer Fundamentals and Office Automation”, BPB Publications, 2019.
4. Marakas, G. M., “Office Automation: Concepts, Tools, and Applications”, Business Publications, Inc., 2018.
5. Juneja, R., & Kaur, K., “Fundamentals of Computers and Office Automation”, Khanna Book Publishing, 2017.

Course Code: CABO–1116

Credit: 2

Course Name: Problem solving Using ‘C’ Lab

Course Objectives:

- To develop students' ability to apply problem-solving techniques using the C programming language.
- To teach fundamental concepts of C programming, including syntax, data types, and control structures.
- To enhance skills in writing, testing, and debugging C programs.
- To provide hands-on experience with algorithms and data structures implemented in C.
- To foster the ability to create efficient and well-structured code to solve computational problems.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental programming concepts, problem-solving techniques, and algorithmic approaches.	L2
CO2	Apply programming constructs and tools to develop programs using control structures, functions, and basic data types.	L3
CO3	Analyze problems and select appropriate algorithms and data structures such as arrays, linked lists, stacks, and queues.	L4
CO4	Evaluate the efficiency and performance of algorithms based on time and space complexity.	L5
CO5	Design and implement programs and projects using algorithms and data structures to solve real-world problems.	L6

Block-I: Programming and Algorithm Lab

Unit 1: Fundamentals of Programming and Problem Solving

- Introduction to programming environments and tools
- Basic programming concepts: variables, data types, control structures
- Algorithmic problem-solving techniques
- Hands-on coding exercises to reinforce programming fundamentals

Unit 2: Advanced Algorithms and Data Structures

- Implementation of sorting and searching algorithms
- Introduction to data structures: arrays, linked lists, stacks, queues
- Algorithmic analysis and efficiency considerations
- Project-based assignments applying algorithms and data structures to real-world problems

References:

1. E. Balagurusamy, “Programming in ANSI C”, 8th Edition, McGraw Hill, 2019.

2. Reema Thareja, “Programming in C”, 2nd Edition, Oxford University Press, 2021.
3. Byron Gottfried, “Programming with C (Schaum’s Outlines Series)”, 3rd Edition, McGraw Hill, 2018.
4. Yashavant Kanetkar, “Let Us C”, 17th Edition, BPB Publications, 2023.
5. Brian W. Kernighan and Dennis M. Ritchie, “The C Programming Language”, 2nd Edition, Pearson Education, 2020.

Semester – II

Course Code: CABO–1201

Credit: 4

Course Name: System Analysis & Design

Course Objectives:

- To equip students with the skills to analyze and design effective information systems.
- To teach fundamental concepts and methodologies of system analysis and design.
- To develop proficiency in using tools and techniques for modeling and documenting systems.
- To enhance understanding of the system development life cycle (SDLC) and project management practices.
- To prepare students to assess and meet organizational needs through well-designed information systems.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of systems, types of information systems, and the role of systems analysts in organizations.	L2
CO2	Apply System Development Life Cycle (SDLC) phases and techniques to analyze and design information systems.	L3
CO3	Analyze business problems and information requirements using tools such as interviews, questionnaires, and structured analysis techniques (DFD, decision trees).	L4
CO4	Evaluate system feasibility, performance, and design alternatives based on economic, technical, and operational criteria.	L5
CO5	Design and develop system models, input-output forms, and maintenance strategies for real-world information systems.	L6

Block-I: System Concepts and Information Systems Environment

Unit-1: The System Concept: Definition, Characteristics of Systems, Elements of a System, Open and Closed System,

Unit-2: Formal and Informal Information Systems, Computer based Information Systems,

Unit-3: Management Information System, Decision Support System, General Business Knowledge, and Interpersonal Communicational System.

Block-II: The System Development Life Cycle

Unit-4: Recognition of needs, Impetus for System Change, Feasibility Study, Analysis, Design, Implementation, Post implementation & Maintenance.

Unit- 5: The Role of the Systems Analyst: Historical Perspective, The War Effort, What Does it takes to do System Analysis, Academic & Personal Qualifications,

Unit-6: The Multi-faceted role of the Analyst, The Analyst/User Interface, Behavior issues.

Block-III: Systems Planning & Initial Investigation

Unit-7: Strategies for Determining Information Requirement.

Unit-8: Problem Definition & Project initiation, Background Analysis, Fact Analysis, Review of Written Documents,

Unit-9: Onsite Observations, Interviews & Questionnaires, Fact Analysis, Performance Analysis, Efficiency Analysis, Service Analysis.

Block-IV: Information Gathering

Unit-10: What Kind of Information do we need? Information about the firms, Information gathering tools,

Unit-11: The art of Interviewing, Arranging the Interview, Guides to a Successful Interview, Types of Interviews and Questionnaires, The Structured and Unstructured Alternatives.

Unit-12: The Tools of Structured Analysis: The Data flow Diagram (DFD), Data Dictionary, Decision Trees and Structured English.

Block-V: Feasibility Study

Unit-13: System performance, Economic Feasibility, Technical Feasibility, Behavioral Feasibility, Steps in Feasibility Analysis.

Unit-14: Input / Output and Forms Design: Input Design, CRT Screen Design, Output Design, Requirements of form Design.

Unit-15: H/W/S/W Selection and Maintenance: The Computer Industry, S/W Industry, a Procedure for H/W/S/W Selection, Major Phases in Selection, Criteria for S/W Selection, The Used Computer, The Computer Contract.

Unit 16: System Maintenance: System Maintenance and Evolution, Maintenance strategies, Software evolution and version control, Change management, Emerging Trends, Introduction to agile methodologies

References:

1. Kendall, K. E., & Kendall, J. E., "Systems Analysis and Design", 11th Edition, Pearson Education, 2023.
2. Satzinger, J. W., Jackson, R. B., & Burd, S. D., "Systems Analysis and Design in a Changing World", 8th Edition, Cengage Learning, 2021.
3. Hoffer, J. A., George, J. F., & Valacich, J. S., "Modern Systems Analysis and Design", 9th Edition, Pearson Education, 2021.
4. Awad, E. M., "Systems Analysis and Design", 2nd Edition, Galgotia Publications, 2019.
5. Alan Dennis, Barbara Haley Wixom, & Roberta M. Roth, "Systems Analysis and Design", 7th Edition, Wiley India, 2020.

Course Code: CABO–1202

Credit: 2

Course Name: C++ & Data Structure

Course Objectives:

- To teach the fundamentals of C++ programming, including syntax, object-oriented principles, and standard libraries.
- To develop skills in designing and implementing data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
- To enhance problem-solving abilities through the application of algorithms and data structures.
- To provide hands-on experience with advanced C++ features like templates, exceptions, and file handling.
- To prepare students to analyze and optimize the efficiency of algorithms and data structures in various computational problems.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain object-oriented programming concepts and fundamental features of C++ including syntax, control structures, arrays, and pointers.	L2
CO2	Apply C++ programming constructs such as classes, functions, inheritance, polymorphism, templates, and exception handling to develop programs.	L3
CO3	Analyze problems and select appropriate data structures such as arrays, linked lists, stacks, queues, and trees for efficient implementation.	L4
CO4	Evaluate the performance of algorithms including sorting, searching, hashing, and graph algorithms based on time and space complexity.	L5
CO5	Design and implement efficient solutions using advanced data structures and algorithms to solve real-world computational problems.	L6

Block I: Introduction to C++ Programming

Unit 1: Introduction to Object-Oriented Programming, Overview of object-oriented programming (OOP) concepts, Understanding classes and objects, Encapsulation, inheritance, and polymorphism

Unit 2: Basics of C++ Programming, Syntax and semantics of the C++ programming language, Data types, variables, and constants, Input and output operations using cout and cin

Unit 3: Control Structures and Functions, Decision-making statements: if-else, switch-CABOe

Looping statements: while, do-while, for, Functions: declaration, definition, and invocation

Unit 4: Arrays and Pointers in C++, Introduction to arrays: declaration, initialization, and accessing elements, Understanding pointers: declaration, arithmetic operations, and indirection, Applications of arrays and pointers in C++ programming

Block II: Advanced C++ Programming Concepts

Unit 5: Classes and Objects in C++, Defining classes and objects in C++, Constructors and destructors

Member functions, access specifiers, and static members

Unit 6: Operator Overloading and Type Conversion, Operator overloading: overloading unary and binary operators, Type conversion operators: conversion constructors and conversion operators, Rules and guidelines for operator overloading and type conversion

Unit 7: Inheritance and Polymorphism, Inheritance: single inheritance, multiple inheritance, and multilevel inheritance, Polymorphism: function overloading and overriding, virtual functions, and dynamic binding, Abstract classes and pure virtual functions

Unit 8: Templates and Exception Handling, Introduction to templates: function templates and class templates, Exception handling: try-catch blocks, throw statement, and exception specifications, Handling exceptions in C++ programs

Block III: Introduction to Data Structures

Unit 9: Introduction to Data Structures, Overview of data structures and their importance in programming, Abstract data types (ADTs) and their implementations, Understanding the concept of efficiency in data structure design

Unit 10: Arrays and Linked Lists, Array-based and linked list-based implementations of lists, Operations on arrays and linked lists: insertion, deletion, searching, and traversal, Comparing the performance of arrays and linked lists

Unit 11: Stacks and Queues, Implementation of stacks and queues using arrays and linked lists, Operations on stacks and queues: push, pop, enqueue, dequeue, Applications of stacks and queues in problem-solving

Unit 12: Trees and Binary Search Trees, Introduction to tree data structures: binary trees, binary search trees (BSTs), Traversing binary trees: preorder, inorder, postorder traversal, Operations on BSTs: insertion, deletion, searching, and balancing

Block IV: Advanced Data Structures and Algorithm Analysis

Unit 13: Graphs and Graph Algorithms, Introduction to graph data structures: directed and undirected graphs, Representation of graphs: adjacency matrix, adjacency list, Graph traversal algorithms: breadth-first search (BFS), depth-first search (DFS)

Unit 14: Sorting and Searching Algorithms, Implementation and analysis of sorting algorithms: bubble sort, selection sort, insertion sort, merge sort, quick sort, Implementation and analysis of searching algorithms: linear search, binary search, Comparing the efficiency of sorting and searching algorithms

Unit 15: Hashing and Hash Tables, Introduction to hashing: hash functions and collision resolution techniques, Implementation of hash tables: chaining, open addressing, Applications of hashing in data storage and retrieval

Unit 16: Advanced Topics in Data Structures, Advanced data structures: AVL trees, B-trees, heap data structures, Algorithm analysis: time complexity, space complexity, Practical applications of data structures and algorithms in solving real-world problems

References:

1. Stroustrup, B., “The C++ Programming Language”, 5th Edition, Addison-Wesley, 2023.
2. Lafore, R., “Data Structures and Algorithms in C++”, 4th Edition, Pearson Education, 2022.
3. Sedgewick, R., & Wayne, K., “Algorithms in C++”, 4th Edition, Addison-Wesley, 2021.
4. Horowitz, E., Sahni, S., & Mehta, D., “Fundamentals of Data Structures in C++”, 2nd Edition, Silicon Press, 2020.
5. Drozdek, A., “Data Structures and Algorithms in C++”, 4th Edition, Cengage Learning, 2020.

Course Code: CABO-1203

Credit: 4

Course Name: Ordinary Differential Equations

Course Objectives:

- To provide a thorough understanding of the fundamental concepts and theories of ordinary differential equations (ODEs).
- To teach methods for solving first-order and higher-order differential equations.
- To develop skills in applying ODE techniques to model and solve real-world problems.
- To enhance proficiency in using analytical and numerical methods for solving ODEs.
- To prepare students for advanced studies in mathematics, engineering, and related fields through a solid foundation in differential equations.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of differential equations including order, degree, formation, and basic solution methods.	L2
CO2	Apply standard techniques such as separation of variables, homogeneous equations, linear and exact methods to solve first-order differential equations.	L3
CO3	Analyze and solve differential equations of higher order and special types including Bernoulli's, Clairaut's, and simultaneous equations.	L4
CO4	Evaluate solutions of differential equations using series methods such as power series and Frobenius method, and interpret special functions like Legendre and Bessel functions.	L5
CO5	Formulate and solve real-world problems by modeling them using differential equations and appropriate solution techniques.	L6

Block I: Differential Equations

Unit-1: Degree, order and solution of a D.E. and Formation of a differential equation

Unit-2: Differential equations of the first order and first degree: Separation of variables method, Solution of homogeneous equations

Unit-3: Linear Differential equations and Bernoulli's Equations

Unit-4: Exact Differential Equations

Block II: Differential equations of the first order but not of the first degree

Unit-5: Differential equations which are solvable for p

Unit-6: Differential equations which are solvable for y

Unit-7: Differential equations which are solvable for x

Unit-8: Clairaut's differential equations

Block III: Higher Order Linear Differential Equation

Unit-9: Rule to find the Complementary function and the Particular Integral.

Unit-10: Cauchy's homogenous linear equation and Legendre's linear equation

Unit-11: Homogeneous linear differential equations

Unit-12: Simultaneous linear differential equations with constant coefficients

Block IV: Series Solutions

Unit-13: Series solutions (Power series, Frobenius method)

Unit-14: Series solutions of second order differential equations

Unit-15: Legendre and Bessel functions (P_n and J_n only) and their properties.

Unit-16: Systems of linear first order ordinary differential equations

References:

1. Barnes, Belinda & Fulford, Glenn R. (2015). Mathematical Modeling with CABOe Studies, Using Maple and MATLAB (3rd ed.). CRC Press, Taylor & Francis Group.
2. Edwards, C. Henry, Penney, David E., & Calvis, David T. (2015). Differential Equation and Boundary Value Problems: Computing and Modeling (5th ed.). Pearson Education.
3. Ross, Shepley L. (2004). Differential Equations (3rd ed.). John Wiley & Sons. India.

Course Code: CABO-1204

Credit: 4

Course Name: Fundamentals of Electronics

Course Objectives:

- To introduce students to the foundational principles and theories of electronics.
- To teach the fundamental concepts of electronic components such as resistors, capacitors, inductors, diodes, and transistors.
- To develop an understanding of basic electronic circuits and their applications.
- To provide hands-on experience with circuit design, construction, and troubleshooting.
- To prepare students for further studies or careers in electronics engineering or related fields.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of electrical quantities, atomic structure, semiconductor materials, and basic circuit components.	L2
CO2	Apply electrical laws (Ohm's and Kirchhoff's) and circuit analysis techniques to solve DC and AC circuit problems.	L3
CO3	Analyze electronic circuits and digital systems using network theorems, Boolean algebra, and logic gate minimization techniques.	L4
CO4	Evaluate the performance and characteristics of semiconductor devices such as diodes, BJTs, FETs, and operational amplifiers.	L5
CO5	Design and develop basic electronic circuits including amplifiers, oscillators, power supplies, and simple digital circuits for practical applications.	L6

Block I: Basic Electrical Concepts and Components

Unit 1: Introduction to Electronics: History and significance of electronics, Nature of atom; Atomic energy levels; Electronic, structure of elements

Unit 2: Electrical Quantities: Voltage, current, resistance, power, and energy, Energy band theory of crystal

Unit 3: Basic Circuit Components: Resistors, capacitors, and inductors, Insulator; Semiconductor and Metals; Mobility and Conductivity

Unit 4: Ohm's Law and Kirchhoff's Laws: Voltage and current relationships in circuits, Donor and Acceptor impurities; Charge density in a semiconductor;

Block II: Circuit Analysis and Techniques & Digital Electronics

Unit 5: Series and Parallel Circuits: Analysis and applications, Network Theorems: Thevenin's, Norton's, and superposition theorems.

Unit 6: AC Fundamentals: Alternating current, RMS values, and phasors, AC Circuit Analysis: Impedance, reactance, and power in AC circuits.

Unit 7: Number system; Conversion of base (decimal, binary, octal and hexadecimal numbers), Addition and Subtraction; Fractional numbers, BCD numbers Boolean algebra, Logic Gates, Concept of Universal Gates

Unit 8: Canonical forms, Minimization using K-map including don't care conditions. Problems based on number systems and Boolean algebra

Block III: Semiconductor Devices

Unit 9: Introduction to Semiconductors: Properties and types of semiconductor materials, Generation and recombination of charges.

Unit 10: Diodes: Characteristics, types, and applications.

Unit 11: Transistors: Bipolar Junction Transistors (BJTs) and their operation.

Unit 12: Field-Effect Transistors (FETs): Types, characteristics, and applications.

Block IV: Basic Electronic Circuits and Applications

Unit 13: Amplifiers: Basic principles, types, and applications, Ideal operational amplifier; Op-Amp parameters and its ideal characteristics

Unit 14: Oscillators: Principles and types of oscillators.

Unit 15: Power Supplies: Rectifiers, filters, and voltage regulators.

Unit 16: Introduction to Digital Electronics: Logic gates, Boolean algebra, and simple digital circuits.

References:

1. Boylestad, R. L., & Nashelsky, L., "Electronic Devices and Circuit Theory", 11th Edition, Pearson Education, 2009.
2. Horowitz, P., & Hill, W., "The Art of Electronics", 3rd Edition, Cambridge University Press, 2015.
3. Scherz, P., & Monk, S., "Practical Electronics for Inventors", 4th Edition, McGraw-Hill Education, 2016.
4. Alexander, C. K., & Sadiku, M. N. O., "Fundamentals of Electric Circuits", 6th Edition, McGraw-Hill Education, 2016.
5. Floyd, T. L., "Electronic Devices (Electron Flow Version)", 10th Edition, Pearson Education, 2017.

Course code: ENBO-1201

Credit: 2

Course Title: Creative Writing

Course Objectives:

- To develop students' creativity and imagination in writing.
- To improve students' writing skills and techniques.
- To help students express themselves effectively through different writing styles and genres.
- To introduce students to various forms of creative writing, such as poetry, short stories, and plays.
- To foster a supportive and collaborative environment for sharing and critiquing each other's work.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts, genres, and processes involved in creative writing.	L2
CO2	Apply writing techniques to develop characters, settings, plots, and dialogues in creative compositions.	L3
CO3	Analyze elements of creative writing such as theme, voice, and symbolism in various literary works.	L4
CO4	Evaluate written content using feedback mechanisms and self-editing techniques to improve quality.	L5
CO5	Create original creative works (stories, narratives, or other genres) incorporating essential elements of effective writing.	L6

Block I: Introduction to Creative Writing

Unit 1: Understanding the Basics of Creative Writing

Unit 2: Exploring Different Genres of Writing

Unit 3: Developing a Writing Routine

Unit 4: Understanding the Importance of Feedback

Unit 5: Practicing Self-editing Techniques

Block II: Elements of Creative Writing

Unit 6: Character Development

Unit 7: Setting and Atmosphere

Unit 8: Plot Development

Unit 9: Dialogue and Voice

Unit 10: Theme and Symbolism

References:

1. Swain, D. V., "Creating Characters: How to Build Story People", 1st Edition, Writer's Digest Books, 2009.
2. Browne, R., & King, D., "Self-Editing for Fiction Writers: How to Edit Yourself into Print", 2nd Edition, HarperCollins, 2004.
3. Bell, J., & Magrs, P. (Eds.), "The Creative Writing Coursebook: Forty Authors Share Advice and Exercises for Fiction and Poetry", 1st Edition, Macmillan, 2001.
4. LaPlante, A., "The Making of a Story: A Norton Guide to Creative Writing", 1st Edition, W. W. Norton & Company, 2007.
5. Rasley, A., "The Power of Point of View: Make Your Story Come to Life", 1st Edition, Writer's Digest Books, 2005.

उद्देश्य

हिंदी के 4Vप, उकार से परिचित कराना।

हिंदी के तकनीकी पद्यों का अध्ययन कराना।

मीडिया लेखन के 4Vप से परिचित कराना।

मीडिया लेखन के विविध Vपों का अध्ययन कराना।

अधिगम की उपलियां

CO1	हिंदी लेखन एवं मीडिया लेखन की मूल अवधारणाओं, 4Vप, उकार तथा महत्त्व को समझना।	L2
CO2	हिंदी के कलात्मक एवं तकनीकी पद्यों (पटकथा, संवाद, चरित्र, उकाश आदि) तथा मीडिया लेखन के विभिन्न Vपों का अनुपयोग करना।	L3
CO3	डिजिटल एवं ऑनलाइन मीडिया के विभिन्न 4Vपों तथा उनके उभाव का विश्लेषण करना।	L4
CO4	हिंदी समीक्षा एवं मीडिया सामग्री की गुणवत्ता, उभावशीलता एवं चुनौतियों का मूलांकन करना।	L5
CO5	हिंदी लेखन एवं मीडिया लेखन के विभिन्न Vपों की रचनात्मक सामग्री (पटकथा, संवाद, समाचार/विज्ञापन लेखन) का सृजन करना।	L6

विद्यार्थी हिंदी लेखन के 4Vप, हिंदी के उकार और हिंदी के तकनीकी पद्यों से परिचित होंगे।

मीडिया लेखन के 4Vप और विविध Vपों का अध्ययन कर सकेंगे।

खंड-1 हिंदी लेखन

इकाई-1 हिंदी लेखन: 4Vप एवं महत्त्व

इकाई-2 हिंदी के उकार

इकाई-3 िफुकाकलाप (पटकथा, संवाद, अभिनय, संगीत)

इकाई-4 डिफ़्यूज़न तकनीकी (डिफ़्यूज़न, उकाश, डिफ़्यूज़न,

संपादन) इकाई-5 डिफ़्यूज़न: 4V एवम् मह

खंड-2 मीडिया लेखन

इकाई-6 मीडिया लेखन: अर्थ, परभाषा एवं अवधारणा

इकाई-7 डिफ़्यूज़न मीडिया के विविध रूप (समाचार-पत्रिकाएँ, पोर्टल,

विज्ञापन) इकाई-8 डिफ़्यूज़न मीडिया के विविध रूप (रेडियो,

टेलीविजन, सोशल मीडिया) इकाई-9 मीडिया लेखन में हिंदी की

भूमिका

इकाई-10 मीडिया लेखन की संभावनाएं एवं चुनौतियां

सहायक सूची

1. कंपटन, ज., "डिफ़्यूज़न और डिफ़्यूज़न की कला", 1^व संस्करण, पठगुइन बुक्स, 2012।
2. बॉडन वेल, ड., और थॉमसन, के., "डिफ़्यूज़न कला: एक परिचय", 10^व संस्करण, मैकग्रा-हिल एजुकेशन, 2018।
3. मण्डल, "आंख की झपकी में डिफ़्यूज़न संपादन पर एक टिप्पणी", 2^व संस्करण, सिलमैन-जेम्स, 2004।
4. गुप्ता, एस., "मीडिया लेखन: संसाधन और प्रौद्योगिकी", 2^व संस्करण, सागर प्रकाशन, 2017।
5. राज, आर., "मीडिया का उभाव: संवाद, रिपोर्टिंग और उचार", 1^व संस्करण, राष्ट्रीय पुस्तकगृह, 2019।

Code: CABO–1205

Credit: 2

Course Name: Logical Reasoning

Course Objectives:

- To equip learners with the ability to critically analyze, interpret, and evaluate arguments and data, enabling them to solve complex problems with accuracy and efficiency.
- To foster the capacity to think clearly and rationally, understanding logical connections between ideas, challenging assumptions, and evaluating evidence.
- To provide learners with the tools to effectively analyze and interpret data presented in various formats, making accurate conclusions and decisions based on this analysis.
- To teach advanced problem-solving techniques, encouraging creative thinking and enabling learners to approach unfamiliar situations and novel problems with confidence.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of logical reasoning including deductive and inductive reasoning, analogies, and logical statements.	L2
CO2	Apply reasoning techniques to solve problems involving Venn diagrams, syllogisms, cubes, and dice.	L3
CO3	Analyze arguments and logical statements with multiple premises using connectives and reasoning patterns.	L4
CO4	Evaluate the validity of arguments by identifying logical fallacies and assessing reasoning accuracy.	L5
CO5	Construct logical arguments and solve complex reasoning problems using appropriate analytical techniques.	L6

Block 1: Foundational Reasoning Skills

Unit 1: Introduction to Logical Reasoning

Unit 2: Deductive Reasoning

Unit 3: Inductive Reasoning

Unit 4: Analogy based on kinds of relationships

Unit 5: Logical statements- Two premise argument, more than two premise argument using connectives.

Block 2: Application of Reasoning

Unit 6: Venn diagrams

Unit 7: Moods and figures

Unit 8: Problem on Cubes and Dices

Unit 9: Syllogism

Unit 10: Logical Fallacies

References:

1. **Morris, J.**, “A Modern Approach to Logical Reasoning”, 4th Edition, McGraw-Hill Education, 2018.
2. **Rao, S. S.**, “Logical and Analytical Reasoning”, 2nd Edition, Tata McGraw-Hill, 2017.
3. **Narayana, B.**, “Mastering Logical Reasoning”, 1st Edition, Pearson Education, 2020.
4. **Kaplan, M.**, “Logical Reasoning for Competitive Exams”, 1st Edition, Kaplan Publishing, 2021.
5. **Khan, M. A.**, “Critical Thinking and Logical Reasoning”, 3rd Edition, Wiley India, 2019.

Course Code: CABO–1216

Credit: 2

Course Name: C++ & Data Structure Lab

Course Objectives:

- To provide practical experience in implementing data structures using C++ programming language.
- To enhance understanding of C++ syntax, object-oriented concepts, and standard libraries through hands-on exercises.
- To develop proficiency in designing, coding, testing, and debugging data structures and algorithms.
- To foster problem-solving skills by applying data structures to solve real-world computational problems.
- To prepare students for software development roles by equipping them with practical skills in C++ programming and data structure implementation.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of data structures and algorithms including sorting, searching, trees, and graphs.	L2
CO2	Apply various sorting and searching techniques to solve computational problems efficiently.	L3
CO3	Analyze and implement data structures such as stacks, queues, linked lists, and trees using static and dynamic memory allocation.	L4
CO4	Evaluate the performance of algorithms such as tree traversals, graph traversal (BFS, DFS), and shortest path techniques based on efficiency.	L5
CO5	Design and develop programs using advanced data structures and algorithms to solve real-world problems.	L6

List of Experiments

Write Programing in C++for following:

Lab 1. Sorting programs: Bubble sort, Merge sort, Insertion sort, Selection sort, and Quick sort.

Lab 2. Searching programs: Linear Search, Binary Search.

Lab 3. Array implementation of Stack, Queue, Circular Queue, Linked List.

Lab 4. Implementation of Stack, Queue, Circular Queue, Linked List using dynamic memory allocation.

Lab 5. Implementation of Binary tree.

Lab 6. Program for Tree Traversals (preorder, in order, post order).

Lab 7. Program for graph traversal (BFS, DFS).

Lab 8. Program for minimum cost spanning tree, shortest path.

References:

1. Schildt, H., “C++: The Complete Reference”, 5th Edition, McGraw-Hill Education, 2014.
2. Tanenbaum, A. S., & Augenstein, M. J., “Data Structures using C++”, 2nd Edition, Pearson Education, 2015.
3. Tanenbaum, A. S., “Data Structures: Algorithms and Applications in C++”, 2nd Edition, Pearson Education, 2017.
4. Balgurusamy, E., “Object-Oriented Programming with C++”, 7th Edition, McGraw-Hill Education, 2017.
5. Yedidyah, L., “Data Structures and Algorithm Analysis in C++”, 4th Edition, Pearson Education, 2018.

Semester - III

Course Code: CABO-1301

Credit: 4

Course Name: Computer Organization

Course Objectives:

- To understand the basic structure, organization, and functional components of computer systems.
- To explain digital logic design, instruction set architecture, and CPU organization.
- To analyze memory hierarchy, cache memory, and input/output organization in computer systems.
- To study modern computer architecture concepts such as pipelining and parallel processing.
- To apply theoretical knowledge through practical assignments and laboratory experiments related to computer hardware and system design.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of computer organization, digital logic, Boolean algebra, and basic components of computer systems.	L2
CO2	Apply principles of digital logic design, sequential circuits, and register transfer operations in computer systems.	L3
CO3	Analyze CPU architecture, memory hierarchy, instruction execution, and input-output organization in computing systems.	L4
CO4	Evaluate system performance, pipelining techniques, cache memory strategies, and parallel processing architectures.	L5
CO5	Design and develop efficient computing systems incorporating modern architectures such as multicore, GPU, and energy-efficient systems.	L6

Block – I: Foundations of Computer Organization

Unit 1: Introduction to Computer Organization and Architecture, Basic concepts and components of computer systems, Historical perspective and evolution of computer architecture, Digital Logic and Circuits, Boolean algebra and logic gates

Unit 2: Digital Logic and Circuits (Continued)

Combinational and sequential circuits, Designing and analyzing digital circuits, Arithmetic circuits and ALU design, Simplification of Boolean functions, NAND or NOR implementation, Don't care condition, Tabulation method, Adder, subtractor, Code Conversion, Universal Gate

Unit 3: Sequential Logic:

Flip-flops, Triggering of Flip-flops, Analysis of clocked sequential, circuits, State reduction and Assignment, Flip-flop excitation

Unit 4: CPU Structure and Function

Instruction set architecture (ISA), CPU components: ALU, control unit, registers, Design of counters, Design with state equations, Register Transfer Language, Register transfer, Bus and Memory transfer

Block – II: Processor Architecture and Design

Unit 5: CPU Structure and Function (Continued), Microarchitecture and instruction execution, Performance metrics and benchmarks

Unit 6: Memory Systems, Memory hierarchy: cache, main memory, secondary storage, Memory technologies and organization

Unit 7: Memory Systems Cache memory: principles and optimization techniques, Virtual memory concepts and implementation.

Unit 8: Basic Computer Organization and Design, Instruction codes, Computer registers, Computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt, Design of Basic computer, Design of Accumulator Unit

Unit 9: Input/Output Systems, I/O interface and devices, Polling, interrupts, and DMA, Programming Arithmetic and logic operations, Subroutines, I-O Programming

Block – III: Memory Systems and Interfacing

Unit 10: Input/Output Systems, Bus architectures and protocols, Storage devices and technologies

Unit 11: Pipelining, Basic concepts and principles of pipelining, Pipelined CPU design and optimization, Pipelining, Hazards and their resolution techniques, Advanced pipelining techniques

Unit 12: Heterogeneous Computing and Accelerators, GPU architectures and programming models, FPGA-based accelerators, Domain-specific accelerators (e.g., AI accelerators)

Unit 13: Parallel Processing, Concepts of parallel processing, Types of parallelism: SIMD, MIMD

Block – IV: Parallel and Distributed Computing

Unit 14: Parallel Processing, Multiprocessor systems and architectures, Cache coherence protocols

Unit 15: Advanced Topics in Computer Architecture, Multicore and manycore architectures, Energy-efficient computing techniques

Unit 16: Emerging Trends in Computer Architecture, Quantum computing, Neuromorphic computing, Energy-Efficient Architectures, Techniques for energy-efficient computing, Low-power processor and system design, Energy-aware programming models and optimizations

References:

1. Morri, M. M., “Computer Organization and Architecture: Designing for Performance”, 10th Edition, Pearson Education, 2019.
2. Carl Hamacher, V., & Zaky, Z., “Computer Organization and Embedded Systems”, 6th Edition, McGraw-Hill Education, 2012.

3. Stallings, W., “Computer Organization and Architecture: Software Perspective”, 9th Edition, Pearson Education, 2015.
4. Hayes, J. P., “Computer Architecture and Organization”, 4th Edition, McGraw-Hill Education, 2012.
5. William, A., “Parallel Computing: Theory and Practice”, 2nd Edition, Pearson Education, 2014.
6. Dube, V. K., & Prabhu, M. M., “Introduction to Computer Architecture and Organization”, 1st Edition, Wiley, 2016.

Course Code: CABO–1302

Credit: 2

Course Name: Operating System

Course Objectives:

- To introduce students to the fundamental concepts and principles of operating systems.
- To teach the structure and components of modern operating systems, including process management, memory management, file systems, and device management.
- To develop an understanding of the role of operating systems in managing hardware resources and providing a platform for software applications.
- To explore advanced topics such as concurrency, synchronization, and deadlock handling.
- To provide hands-on experience with operating system concepts through practical exercises, simulations, and projects.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of operating systems including structure, services, architecture, and interfaces.	L2
CO2	Apply process management techniques such as scheduling algorithms, synchronization, and inter-process communication mechanisms.	L3
CO3	Analyze memory management strategies including paging, segmentation, virtual memory, and page replacement algorithms.	L4
CO4	Evaluate file system structures, disk scheduling algorithms, and I/O management techniques for system performance and reliability.	L5
CO5	Design and implement efficient operating system components or solutions incorporating process, memory, file, and I/O management concepts.	L6

Block I: Introduction to Operating Systems

Unit 1: Fundamentals of Operating Systems: Definition and objectives of operating systems, History and evolution of operating systems, Types of operating systems (e.g., batch processing, time-sharing, real-time), Operating system structures

Unit 2: Operating System Services: Process management, Memory management, File system management, Device management, System calls and APIs

Unit 3: Operating System Architecture: Kernel architecture, User mode and kernel mode, System components (e.g., scheduler, dispatcher, interrupt handler), Microkernel and monolithic kernel architectures

Unit 4: Operating System Interfaces: Command-line interfaces (CLI), Graphical user interfaces (GUI), Application programming interfaces (APIs), Shell scripting

Block II: Process Management

Unit 5: Introduction to Processes: Definition of a process, Process states and transitions, Process control block (PCB), Process creation and termination, Process hierarchy

Unit 6: Process Scheduling: Basics of process scheduling, CPU scheduling algorithms (e.g., FCFS, SJF, Round Robin, Priority Scheduling), Multilevel Queue Scheduling, Thread scheduling, Scheduling in multiprocessor systems

Unit 7: Process Synchronization: Basics of process synchronization, Critical section problem, Semaphores and mutex locks, Monitors, Deadlock prevention and avoidance

Unit 8: Inter-process Communication: Inter-process communication mechanisms (e.g., shared memory, message passing), Pipes and FIFOs, Socket programming, Remote procedure calls (RPC), CAbOe studies and applications of inter process communication

Block III: Memory Management

Unit 9: Introduction to Memory Management, Basic concepts of memory management, Memory hierarchy (registers, cache, main memory, secondary memory), Address spaces and memory allocation

Unit 10: Memory Allocation Techniques Fixed partitioning, Dynamic partitioning, Buddy system, Paging and segmentation, Virtual memory concepts

Unit 11: Memory Management Strategies, Swapping and swapping strategies, Page replacement algorithms (e.g., FIFO, LRU, Optimal), Segment replacement algorithms, Thrashing and its prevention

Unit 12: Advanced Memory Management Topics, Memory fragmentation, Memory compaction, Memory protection and access control, Shared memory and memory-mapped files

Block IV: File System and Input/Output Management

Unit 13: Introduction to File Systems, Basic concepts of file systems, File system objectives and functions, File system types (e.g., FAT, NTFS, ext4), File system architecture

Unit 14: File System Implementation, File organization and structures, Directory structures and operations, File system operations (e.g., create, delete, read, write), File system consistency and recovery

Unit 15: Input/Output Management, I/O devices and device controllers, I/O operations and techniques (e.g., polling, interrupts, DMA), Buffering and caching, Device drivers and I/O software layers

Unit 16: Advanced Topics in File Systems and I/O Management, File system security and access control, Disk scheduling algorithms, RAID (Redundant Array of Independent Disks), I/O performance optimization techniques

References:

1. Silberschatz, A., Galvin, P. B., & Gagne, G., "Operating System Concepts", 10th Edition, Wiley, 2018.
2. Tanenbaum, A. S., & Bos, H., "Modern Operating Systems", 4th Edition, Pearson Education, 2015.

3. Stallings, W., "Operating Systems: Internals and Design Principles", 9th Edition, Pearson Education, 2017.
4. Pratt, M. S., & Zeldman, A. S., "Operating Systems: A Design-Oriented Approach", 2nd Edition, McGraw-Hill Education, 2016.
5. Dhamdhere, D. M., "Operating Systems: A Concept-Based Approach", 2nd Edition, Tata McGraw-Hill, 2015.
6. McHoes, A. D., & Irvine, J. E., "Operating Systems: Internals and Design Principles", 8th Edition, Pearson Education, 2017.

Course Code: CABO-1303

Credit: 4

Course Name: Integral Calculus

Course Objectives:

- To provide a comprehensive understanding of integral calculus concepts, including antiderivatives, definite integrals, and the fundamental theorem of calculus.
- To teach various integration techniques, such as substitution, integration by parts, and trigonometric substitution.
- To develop proficiency in calculating areas, volumes, and other geometric quantities using integration.
- To explore applications of integral calculus in physics, engineering, economics, and other disciplines.
- To prepare students for further studies in mathematics or related fields by mastering integral calculus techniques and applications.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of Riemann integration, improper integrals, and vector calculus including their definitions and properties.	L2
CO2	Apply techniques of integration, convergence tests, and vector differentiation to solve mathematical problems.	L3
CO3	Analyze integrals and vector fields using theorems such as mean value theorem, Green's, Gauss, and Stokes' theorem.	L4
CO4	Evaluate the convergence of improper integrals and compute areas, volumes, and surface integrals in various coordinate systems.	L5
CO5	Formulate and solve real-world problems using advanced integration techniques and vector calculus concepts.	L6

Block I: Riemann Integration

Unit 1: Partition on an Interval, Riemann upper and lower sums

Unit 2: Criterion and properties of Riemann integration

Unit 3: Fundamental and mean value theorems of integral calculus

Unit 4: Integration as a limit of sum and Differentiation under the sign of Integration

Block II: Improper Integral

Unit 5: Improper integrals and their classification

Unit 6: Test for convergence-I

Unit 7: Test for convergence-II

Unit 8: Beta and Gamma functions

Block III: Rectification and Quadrature, Applications of Integration Unit 9: Reduction formulae

Unit 10: Rectification and Quadrature

Unit 11: Volumes and surfaces of solids of revolution

Unit 12: Double and triple integrals

Block IV: Vector Differentiation and Integration

Unit 13: Vector Differentiation: Gradient, Divergence and Curl

Unit 14: Normal on a surface, Directional Derivative

Unit 15: Vector Integration

Unit 16: Green, Gauss, Stokes Theorem and their applications

References:

1. Thomas, G. B., & Finney, R. L., “Calculus and Analytical Geometry”, 9th Edition, Pearson Education, 2015.
2. Stewart, J., “Calculus: Early Transcendentals”, 8th Edition, Cengage Learning, 2015.
3. Anton, H., Bivens, I., & Davis, S., “Calculus: Multivariable”, 10th Edition, Wiley, 2019.
4. Morris, S. A., & Lardner, J. W., “Calculus: An Introduction to the Theory of Functions of One and Several Variables”, 1st Edition, Springer, 2014.
5. Ross, S. L., “Elementary Analysis: The Theory of Calculus”, 2nd Edition, Springer, 2017.

Course code: ENBO-1301

Credit: 2

Course title: Personality Development

Course Objectives:

- To provide participants with a comprehensive framework for personal growth and development. By the end of the course, participants will:
- To gain a deep understanding of the various components that make up an individual's personality, including temperament, character, and traits.
- To develop heightened self-awareness regarding one's strengths, weaknesses, emotions, thoughts, and values.
- To enhance verbal and non-verbal communication skills to interact more effectively with others in personal and professional settings.
- To improve emotional intelligence by learning to manage and express one's emotions constructively and understand the emotions of others.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain key concepts of personality development, communication skills, emotional intelligence, time management, and stress management.	L2
CO2	Apply effective communication, time management, and stress management techniques in personal and professional contexts.	L3
CO3	Analyze personal strengths, interpersonal relationships, and problem-solving approaches to improve effectiveness.	L4
CO4	Evaluate leadership qualities, teamwork dynamics, and decision-making strategies in various situations.	L5
CO5	Develop a personal growth plan incorporating critical thinking, adaptability, resilience, and career planning strategies.	L6

Block I: Understanding Self and Interpersonal Skills

Unit 1: Introduction to Personality Development

Unit 2: Communication Skills

Unit 3: Emotional Intelligence

Unit 4: Time Management

Unit 5: Stress Management

Block II: Enhancing Personal Effectiveness and Building Relationships

Unit 6: Critical Thinking and Problem Solving

Unit 7: Leadership and Teamwork

Unit 8: Adaptability and Resilience

Unit 9: Personal Branding and Networking

Unit 10: Planning for the Future

References:

1. Covey, Stephen R., "The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change," Simon & Schuster, 1989.
2. DeVito, Joseph A., "The Interpersonal Communication Book," Pearson, 2015.
3. Goleman, Daniel, "Emotional Intelligence: Why It Can Matter More Than IQ," Bantam Books, 1995.
4. Kouzes, James M., & Posner, Barry Z., "The Leadership Challenge: How to Make Extraordinary Things Happen in Organizations," Jossey-Bass, 2017.

उहे4

पटकथा लेखन का पुरचय कराना।

विभाथियों की लेखन मता का विकास

विभाथियों को लेखन रोजगार संबधी कों के लिए तैयार करना।

अधगमकीउपलियां

CO1	पटकथा लेखन की मूल अवधारणाओं, तों, ँकारों एवं संरचना (ी ए4 एवं फाइव ए4) को समझाना।	L2
CO2	पटकथा लेखन की तकनीको एवं शवावली का उपयोग करते 5ए विभष ँकार की पटकथाओं का निमानण करना।	L3
CO3	पटकथा लेखन रशोध, संरचना एवं विभष माओं (वेब सीरीज़, लघु िफु डुमकी, वि्यापन) की आवकताओं का विषण करना।	L4
CO4	विभष पटकथाओं की गुणवका, संरचना एवं ँभावशीलता का मूांकन करना।	L5
CO5	वेब सीरीज़, लघु िफु डुमकी एवं वि्यापन के लिए ँभावी और रचनाक पटकथा का सृजन करना।	L6

पटकथा-लेखन के 4Vप और तकनीकी शों से परिचत होंगे।

पटकथा लेखन की जानकारी मिलने के उपरांत विभाथा रोजगार-बाज़ार के लिए तैयार

होंगे। अभि4 कौ शलका विकास।

भाषायी संषण के महे से अवगत होंगे।

खंड-1पटकथालेखन

इकाई-1 पटकथा-लेखन:4Vप, ते

इकाई-2 पटकथा लेखन के ँकार

इकाई-3 पटकथा लेखन संशोधन का महत्त्व

इकाई-4 पटकथा की शैली तथा

एक अंक (एक) और पांच अंक (पाँच) को समझना

खंड-2 पटकथालेखन-2

इकाई-5 वेबसीरीज़के लिए पटकथालेखन

इकाई-6 लघु फिल्मके लिए पटकथा

लेखन इकाई-7 डॉक्यूमेंट्रीके लिए

पटकथा लेखन

इकाई-8 विज्ञापन फिल्मके लिए पटकथा लेखन

सहायकपुस्तकें

1. कथासंरचना – काशीनाथ सिंह
संरक्षण: प्रथम, 2015 प्रकाशक:
राजकमल प्रकाशन
2. पटकथा लेखन – रामेश्वर प्रसाद
संरक्षण: द्वितीय, 2017
प्रकाशक: यज्ञकोश प्रकाशन
3. सिनेमाई लेखन – प्रदीप कुमार
संरक्षण: प्रथम, 2018 प्रकाशक:
सागर प्रकाशन
4. चित्रपट लेखन कला – सुरेश
चतुर्वेदी संरक्षण: प्रथम, 2016
प्रकाशक: हिंदी प्रकाशन मंडल
5. टीवी और फिल्मके लिए पटकथा लेखन – सुभाष
शर्मा संरक्षण: प्रथम, 2019
प्रकाशक: अक्षय प्रकाशन
6. पटकथा लेखन: सिनेमांत और अंतरास – विश्व
कुमार संरक्षण: प्रथम, 2020
प्रकाशक: भारतीय साहित्य प्रकाशन

Course code: VACO-1301

Credit: 3

Course Title: Environmental Education

Course Objectives:

- To learn about the Earth's natural systems, including ecosystems, biodiversity, and the processes that support life. They will explore the interconnections between these systems and human societies.
- To introduce to global, regional, and local environmental challenges, including pollution, resource depletion, and biodiversity loss, understanding their causes and effects.
- To equip students with the knowledge and skills to develop and evaluate sustainable solutions to environmental challenges, emphasizing the role of innovation and technology.
- To reflect on their roles and responsibilities in mitigating environmental issues, promoting a sense of stewardship towards the planet.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the concepts of natural resources, ecosystems, biodiversity, and environmental pollution.	L2
CO2	Apply knowledge of resource management, waste management, and energy conservation in real-life situations.	L3
CO3	Analyze the impact of human activities on ecosystems, biodiversity, and environmental sustainability.	L4
CO4	Evaluate environmental policies, conservation strategies, and sustainable industrial practices.	L5
CO5	Develop solutions and strategies for environmental protection and sustainable development.	L6

Block I: Understanding Natural Resources

Unit 1: Introduction to Natural Resources

Unit 2: Water Resources

Unit 3: Soil Resources

Unit 4: Forest Resources, Forest management and conservation

Unit 5: Mineral and Energy Resources

Block II: Ecosystems and Biodiversity

Unit 6: Basics of Ecology and Ecosystems

Unit 7: Terrestrial Ecosystems

Unit 8: Aquatic Ecosystems

Unit 9: Urban Ecosystems

Unit 10: Global Biodiversity Hotspots

Block III: Pollution and its prevention

Unit 11: Pollution: Meaning and types

Unit 12: Solid Waste Management

Unit 13: Sustainable Practices in Industries

Unit 14: Energy Conservation and Renewable Energies

Unit 15: Environmental Policies and Legislation

References:

1. Miller, G. Tyler, and Scott Spoolman. "Living in the Environment." Cengage Learning, 18th edition, 2015.
2. Chiras, Daniel D. "Environmental Science." Jones & Bartlett Learning, 9th edition, 2013.
3. Chapin III, F. Stuart, Pamela A. Matson, and Peter Vitousek. "Principles of Terrestrial Ecosystem Ecology." Springer, 2011.
4. Gaston, Kevin J., and John I. Spicer. "Biodiversity: An Introduction." Blackwell Science, 2nd edition, 2004.
5. Raven, Peter H., Linda R. Berg, and David M. Hassenzahl. "Environment." Wiley, 8th edition, 2011.
6. McKinney, Michael L., Robert M. Schoch, and Logan Yonavjak. "Environmental Science: Systems and Solutions." Jones & Bartlett Learning, 5th edition, 2012.

Course Code: CABO-1304

Credit: 3

Course Name: Computer Based Numerical Analysis

Course Objectives:

- To introduce students to numerical methods for solving mathematical problems using computers.
- To teach algorithms and techniques for numerical approximation, including interpolation, differentiation, and integration.
- To develop skills in error analysis and numerical stability to ensure accurate results.
- To explore applications of numerical analysis in solving problems from various fields such as engineering, physics, and finance.
- To provide hands-on experience with implementing numerical algorithms using computational software or programming languages.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of numerical analysis including error analysis, interpolation, and numerical methods.	L2
CO2	Apply numerical methods such as bisection, Newton-Raphson, interpolation formulas, and numerical integration techniques to solve problems.	L3
CO3	Analyze numerical techniques for solving algebraic equations, linear systems, and ordinary differential equations.	L4
CO4	Evaluate the accuracy, stability, and efficiency of numerical methods based on error analysis and convergence criteria.	L5
CO5	Develop computational solutions using appropriate numerical methods for real-world mathematical and engineering problems.	L6

Block I: Error Analysis and Solution of Algebraic and Transcendental Equation

Unit 1: Accuracy of numbers and Error Analysis

Unit 2: Bisection Method, Iteration method

Unit 3: Method of false position,

Unit 4: Method of finding complex roots: Newton-Raphson method

Block II: Interpolation and Approximation

Unit 5: Finite Differences and Difference table

Unit 6: Newton's forward and backward formula

Unit 7: Gauss' forward and backward interpolation formula, Stirling's formula

Unit 8: Interpolation with Unequal Intervals

Block III: Differentiation and Integration

Unit 9: Numerical Differentiation

Unit 10: Newton Cote's Quadrature Formula and Trapezoidal rule

Unit 11: Simpson's 1/3 and 3/8 rule

Unit 12: Boole's rule and Waddle's rule.

Block IV: Numerical Solution of Linear Algebraic Equation and Ordinary Differential Equations

Unit 13: Gauss-Elimination Method, Gauss-Jordan Method

Unit 14: LU Decomposition Method

Unit 15: Jacobi's Method and Gauss-Seidel Method

Unit 16: Numerical Solution of Ordinary Differential Equations

References:

1. Ranganatham, S., Prasad, M. V. S. S. N., & Ramesh Babu, V., *Numerical Analysis*, S. Chand Publishing, 2010.
2. Jain, M. K., Iyengar, S. R. K., & Jain, R. K., *Numerical Methods*, New Age International, 2007.
3. Sastry, S. S., *Introductory Methods of Numerical Analysis*, PHI Learning Pvt. Ltd., 2012.
4. Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill Education, 2002.
5. Gupta, R. S., *Elements of Numerical Analysis: Algebra*, Cambridge University Press, 2004.
6. Conte, S., & Deboor, C., *Elementary Numerical Analysis*, McGraw-Hill Education, 2013.

Course Code: CABO-1315

Credit: 2

Course Name: Operating System Lab

Course Objective:

- To provide practical experience in working with operating system concepts and functionalities.
- To familiarize students with various operating system commands, utilities, and tools.
- To develop proficiency in system administration tasks such as user management, file system operations, and process management.
- To offer hands-on exercises to reinforce theoretical concepts learned in the operating systems course.
- To prepare students for troubleshooting common operating system issues and challenges.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of operating system processes, scheduling, synchronization, memory management, and file systems.	L2
CO2	Apply system calls and implement programs for process management, inter-process communication, and file operations.	L3
CO3	Analyze CPU scheduling algorithms, synchronization mechanisms, and memory management techniques through simulation.	L4
CO4	Evaluate the performance of algorithms such as scheduling, page replacement, disk scheduling, and deadlock handling techniques.	L5
CO5	Design and develop system-level programs to solve real-world problems involving process synchronization, memory allocation, and resource management.	L6

List of Experiments

Lab 1: Implement a program to create and terminate processes using system calls.

Lab 2: Implement inter-process communication mechanisms such as shared memory and message passing.

Lab 3: Implement CPU scheduling algorithms (e.g., FCFS, SJF, Round Robin) and analyze their performance using simulation.

Lab 4: Implement priority-based CPU scheduling and compare it with other scheduling algorithms.

Lab 5: Implement semaphores for process synchronization and solve synchronization problems like producer-consumer and reader-writer.

Lab 6: Implement mutexes and condition variables for synchronization and demonstrate their usage.

Lab 7: Implement deadlock detection algorithms such as Banker's algorithm and demonstrate their effectiveness.

Lab 8: Implement deadlock prevention techniques like resource allocation graphs and demonstrate their application.

Lab 9: Implement memory allocation algorithms (e.g., contiguous allocation, paging) and analyze their efficiency.

Lab 10: Simulate memory allocation and deallocation to analyze fragmentation issues and mitigation strategies.

Lab 11: Implement demand paging and page fault handling algorithms.

Lab 12: Implement page replacement algorithms (e.g., LRU, FIFO) and evaluate their performance through simulation.

Lab 13: Implement file creation, deletion, and manipulation operations using system calls.

Lab 14: Implement directory operations such as creation, listing, and deletion.

Lab 15: Implement disk scheduling algorithms (e.g., FCFS, SSTF, SCAN) and compare their performance.

Lab 16: Implement input/output operations and device management using system calls.

References:

1. Silberschatz, A., Galvin, P. B., & Gagne, G., *Operating System Concepts*, 10th Edition, Wiley, 2018.
2. Tanenbaum, A. S., & Bos, H., *Modern Operating Systems*, 5th Edition, Pearson Education, 2022.
3. Stallings, W., *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson Education, 2018.
4. Nutt, G. J., *Operating Systems: A Modern Perspective*, 2nd Edition, Addison-Wesley, 2004.
5. Bovet, D. P., & Cesati, M., *Understanding the Linux Kernel*, 3rd Edition, O'Reilly Media, 2005.

Semester – IV

Course Code: CABO-1401

Credit: 4

Course Name: Analysis and Design of Algorithm

Course Objectives:

- To introduce students to fundamental algorithms and algorithm design techniques.
- To teach methods for analyzing the efficiency and correctness of algorithms.
- To develop skills in designing and implementing efficient algorithms to solve computational problems.
- To explore advanced topics such as dynamic programming, greedy algorithms, and divide-and-conquer strategies.
- To provide practical experience through problem-solving exercises, algorithmic challenges, and programming assignments.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental concepts of algorithms, problem-solving approaches, and asymptotic analysis including time and space complexity.	L2
CO2	Apply algorithm design paradigms such as divide and conquer, dynamic programming, and greedy methods to solve computational problems.	L3
CO3	Analyze the efficiency and correctness of algorithms using asymptotic notations and compare different algorithmic strategies.	L4
CO4	Evaluate advanced algorithms including graph algorithms, network flow, string matching, and approximation techniques based on performance and feasibility.	L5
CO5	Design and develop optimized solutions for complex problems using advanced paradigms such as backtracking, branch and bound, randomized and parallel algorithms.	L6

Block I: Introduction to Algorithms and Analysis

Unit 1: Introduction to Algorithms: Definition of algorithms, algorithmic problem-solving, and algorithm design paradigms, Overview of algorithm analysis techniques: time complexity, space complexity, and big-O notation.

Unit 2: Algorithm Analysis: Asymptotic analysis: big-O, big-Theta, and big-Omega notation, Worst-CABOe, average-CABOe, and best-CABOe analysis of algorithms.

Unit 3: Divide and Conquer: Divide and conquer paradigm: principles and applications, Examples of divide and conquer algorithms: merge sort, quicksort, binary search.

Unit 4: Advanced Data Structures:: Priority queues and heaps, Disjoint-set data structures, Hash tables and hashing techniques.

Block II: Advanced Algorithm Design Paradigms

Unit 5: Dynamic Programming: Principles of dynamic programming and memorization, Examples of dynamic programming algorithms: Fibonacci sequence, knapsack problem.

Unit 6: Greedy Algorithms: Greedy algorithm paradigm: characteristics and applications, Examples of greedy algorithms: minimum spanning tree, shortest path algorithms.

Unit 7: Backtracking: Backtracking paradigm: principles and techniques, Examples of backtracking algorithms: N-Queens problem, Sudoku solver.

Unit 8: Branch and Bound: General method, applications - travelling sales person problem, knapsack problem- LC branch and bound solution, FIFO branch and bound solution

Block III: Specialized Topics in Algorithm Design

Unit 9: Graph Algorithms: Graph representation and traversal techniques: DFS, BFS,

Unit 10: Path Algorithms: Shortest path algorithms (Dijkstra's, Bellman-Ford), minimum spanning tree algorithms (Prim's, Kruskal's).

Unit 11: Network Flow Algorithms Maximum flow and minimum cut problems, Network flow algorithms: Ford-Fulkerson algorithm, Edmonds-Karp algorithm.

Unit 12: String Matching Algorithms Brute-force and efficient string-matching algorithms. Examples: Knuth-Morris-Pratt (KMP) algorithm, Boyer-Moore algorithm.

Unit 13: Approximation Algorithms: Introduction to approximation algorithms and approximation ratio. Examples: vertex cover, traveling salesman problem.

Block IV: Randomization Algorithm and Advanced Algorithms

Unit 14: Randomized Algorithms: Randomized algorithm paradigm: principles and applications Examples: randomized quicksort, Monte Carlo algorithms.

Unit 15: Parallel and Distributed Algorithms: Parallel algorithm design techniques: parallel prefix, parallel sorting, Distributed algorithm paradigms: message-passing, shared memory.

Unit 16: Advanced Topics: NP-HARD AND NP-COMPLETE PROBLEMS: Basic concepts, non-deterministic algorithms, NP-hard and NP-complete classes, Cook's theorem.

References:

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 4th ed., MIT Press, 2022.
2. Kleinberg, Jon, and Éva Tardos. *Algorithm Design*. 1st ed., Pearson, 2005.
3. Dasgupta, Sanjoy, Christos H. Papadimitriou, and Umesh Vazirani. *Algorithms*. 1st ed., McGraw-Hill Education, 2006.
4. Sedgewick, Robert, and Kevin Wayne. *Algorithms*. 4th ed., Addison-Wesley Professional, 2011.
5. Skiena, Steven S. *The Algorithm Design Manual*. 3rd ed., Springer, 2020.

Course Code: CABO-1402

Credit: 2

Course Name: Object Oriented Programming with JAVA

Course Objectives:

- To introduce students to the principles and concepts of object-oriented programming (OOP) using the Java programming language.
- To teach the fundamentals of Java syntax, data types, control structures, and object-oriented principles such as encapsulation, inheritance, and polymorphism.
- To develop proficiency in designing, implementing, and testing Java programs using object-oriented design principles.
- To explore advanced Java features and libraries for building robust and scalable applications.
- To provide hands-on experience with real-world Java programming projects to reinforce learning objectives

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of object-oriented programming and Java syntax, including classes, objects, data types, and control structures.	L2
CO2	Apply object-oriented principles such as inheritance, polymorphism, abstraction, and encapsulation to develop Java programs.	L3
CO3	Analyze Java programs involving exception handling, file I/O, generics, and multithreading to ensure correctness and efficiency.	L4
CO4	Evaluate GUI-based applications, event handling mechanisms, and web components (Servlets and JSP) for usability and performance.	L5
CO5	Design and develop complete Java applications, including GUI interfaces, multithreaded systems, and dynamic web applications using Servlets and JSP.	L6

Block I: Introduction to Java Programming

Unit 1: Introduction to Object-Oriented Programming, Concepts of OOP: classes, objects, inheritance, polymorphism, encapsulation, Introduction to Java programming language: history, features, and syntax.

Unit 2: Java Basics, Data types, variables, and operators in Java, Control flow statements: if-else, switch, loops (while, do-while, for).

Unit 3: Classes and Objects, Declaring classes and creating objects, Class members: fields, methods, constructors.

Block II: Object-Oriented Programming Concepts

Unit 4: Inheritance and Polymorphism, Inheritance: extending classes, superclass-subclass relationship, Polymorphism: method overriding, dynamic method dispatch.

Unit 5: Abstraction and Encapsulation, Abstraction: abstract classes, interfaces, and abstract methods, Encapsulation: access modifiers, getter and setter methods.

Unit 6: Packages and Access Modifiers, Package concept in Java: organizing classes into packages, Access modifiers: public, private, protected, default.

Block III: Advanced Java Programming

Unit 7: Exception Handling, Handling exceptions in Java: try-catch, throw, throws, finally blocks, Exception hierarchy and checked vs. unchecked exceptions.

Unit 8: Generics, Introduction to generics: parameterized classes, methods, and interfaces, Type safety and generic collections (ArrayList, HashMap).

Unit 9: File Handling and I/O, File handling in Java: reading from and writing to files, Input/Output streams: byte stream and character stream classes.

Block IV: Event and GUI programming

Unit 10: GUI Programming Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout,

Unit 11: GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing

Unit 12: I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random, Access Files.

Unit 13: Multithreading: Introduction to multithreading: creating and managing threads, Synchronization: synchronized methods, blocks, and thread safety.

Block-IV: Java Servlets & Java Server Pages

Unit-14: Introduction, HTTP Servlet Basics, The Servlet Lifecycle, Retrieving Information, Sending HTML Information, Session Tracking, Database Connectivity

Unit-15: Introducing Java Server Pages, JSP Overview, Setting Up the JSP Environment

Unit-16: Generating Dynamic Content, Using Custom Tag Libraries and the JSP Standard Tag Library, Processing Input and Output.

References:

1. Deitel, Paul, and Harvey Deitel. *Java: How to Program, 11th Edition (Early Objects)*, Pearson, 2017.
2. Savitch, Walter, and Kenrick Mock. *Java: An Introduction to Problem Solving and Programming, 8th Edition*, Pearson, 2021.
3. Eckel, Bruce. *Thinking in Java, 4th Edition*, Prentice Hall, 2006.
4. Horstmann, Cay S. *Core Java Volume I—Fundamentals, 11th Edition*, Prentice Hall, 2018.
5. Bloch, Joshua. *Effective Java, 3rd Edition*, Addison-Wesley Professional, 2018.

Course Code: CABO-1403

Credit: 4

Course Name: Matrix Theory

Course Objectives:

- To provide a comprehensive understanding of the fundamental concepts and properties of matrices.
- To teach methods for performing matrix operations such as addition, multiplication, and inversion.
- To explore applications of matrices in various mathematical and scientific fields, including linear algebra, graph theory, and computer graphics.
- To develop proficiency in solving systems of linear equations using matrix techniques.
- To prepare students for advanced studies in mathematics, engineering, or other disciplines requiring matrix theory skills.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of matrices, determinants, matrix operations, and their basic properties.	L2
CO2	Apply matrix operations, rank determination, and methods such as Cramer's rule and inverse techniques to solve systems of linear equations.	L3
CO3	Analyze matrix structures using concepts like rank, echelon form, eigenvalues, and eigenvectors to understand system behavior.	L4
CO4	Evaluate properties and applications of matrices using theorems such as Cayley–Hamilton theorem and diagonalization techniques.	L5
CO5	Design and represent mathematical models using matrices, quadratic forms, and bilinear forms for solving real-world and computational problems.	L6

Block I: Matrices and their basic properties

Unit 1: Matrices and their types, Related terms

Unit 2: Addition and Multiplication of Matrices

Unit 3: Determinant of a matrix,

Unit 4: Invertible Matrix and their applications.

Block II: Rank and system of linear equations

Unit 5: Rank of a Matrix, Elementary row transformation and echelon form.

Unit 6: Normal form of a matrix, Inverse of a matrix by elementary row transformations.

Unit 7: Consistent and Inconsistent System of linear equations

Unit 8: Cramer's rule

Block III: Eigenvalues and Eigenvectors

Unit 9: Eigen values and Eigenvectors of a square matrix

Unit 10: Basic Properties of Eigen values and Eigenvectors

Unit 11: Cayley's Hamilton Theorem and applications

Unit 12: Diagonalization of matrices

Block IV: Quadratic and Bilinear form

Unit 13: Introduction to Quadratic forms

Unit 14: Matrix representation of Quadratic forms

Unit 15: Diagonalization and Canonical forms

Unit 16: Bilinear forms

References:

1. Zhang, Fuzhen. *Matrix Theory: Basic Results and Techniques*, 2nd Edition, Springer, 2011.
2. Abadir, Karim M., and Jan R. Magnus. *Matrix Algebra*, 1st Edition, Cambridge University Press, 2005.
3. Narayan, Shanti, and P. K. Mittal. *A Textbook of Matrices*, Revised Edition, S. Chand Publishing, 2010.
4. Singh, Arindama. *Introduction to Matrix Theory*, Springer, 2021.
5. Datta, Kanti Bhushan. *Matrix and Linear Algebra Aided with MATLAB*, 2nd Edition, PHI Learning Pvt. Ltd., 2016.
6. Serre, Denis. *What Are Matrices?*, Springer, 2010

Course Code: CABO-1404

Credit: 3

Course Name: Database Management System

Course Objectives:

- To introduce students to the fundamental concepts and principles of database management systems (DBMS).
- To teach database design methodologies, including entity-relationship modeling and normalization.
- To develop proficiency in using SQL (Structured Query Language) for data definition, manipulation, and retrieval.
- To explore advanced topics such as transaction management, concurrency control, and database security.
- To provide hands-on experience with designing, implementing, and querying databases using a DBMS platform.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of database systems, data models, DBMS architecture, E-R modeling, and relational database principles.	L2
CO2	Apply normalization techniques, functional dependencies, transaction management, and SQL queries to design and manipulate databases.	L3
CO3	Analyze database schemas, concurrency control mechanisms, and query processing techniques to ensure data consistency and efficiency.	L4
CO4	Evaluate database performance, indexing methods, transaction schedules, and security mechanisms based on reliability and scalability.	L5
CO5	Design and develop efficient database systems using advanced concepts such as indexing (B/B+ trees), file organization, and modern DBMS technologies.	L6

Block -I: Overview of Database Management System

Unit-1: Introduction, Elements of Database System, DBMS and its architecture, Advantage of DBMS, Types of database users, Role of Database administrator, Need of DBMS.

Unit-2: Brief overview of Hierarchical Model, Network Model, Detailed study of Relational Model (Relations, Properties, Key & Integrity rules),

Unit-3: Comparison of Hierarchical, Network and Relational Model, Data Abstraction & data Independence, Primary Key, Foreign Key, all types of Keys

Unit-4: CODD's rules for Relational Model, E-R diagram, View, Structure.

Block- II: Normalization

Unit-5: Normalization concepts and update anomalies, Functional dependencies, Multi valued and join dependencies.

Unit-6: Normal Forms: (1 NF, 2 NF, 3NF, BCNF, 4NF, and 5NF)

Unit-7: ACID properties, Schedule, Conflict, Conflict Equivalent Schedules, Conflict Serializable Schedule, View Serializable Schedule, Concurrency Problems

Unit-8: Transactions Management & Concurrency Control, Transactions, Recoverable and Irrecoverable Schedule, CABOading and CABOcade less Schedule, Simple Lock based Protocol, Basic 2PL Locking Protocol, 2 PL, 2 PL categories

Block- III: SQL

Unit-9: SQL Constructs, Constraint, Types, Integrity

Unit-10: SQL Join: Multiple Table Queries, Alter v/s Update, Delete v/s Drop v/s Truncate, Constraints in SQL, Aggregate Function, "Group By" Clause, "Having" Clause, "Order By" Clause, Nested and Correlated Nested Query

Unit-11: Build-in functions, Views and their use, WITH Clause, ANY & ALL Operators, IN & NOT IN Operators, EXISTS and NOT EXIST Operators, Set Operations

Unit-12: Overviews of ORACLE: (Data definition and manipulation)

Block- IV: File Organization & Database Security

Unit-13: Security and Integrity threats, Defense mechanism, Integrity, Auditing and Control,

Unit-14: Recent trends in DBMS- Distributed and Deductive Database, Temporal, Geospatial Databases, Multimedia databases

Unit-15: File Organization, Indexing //Dense Index and Sparse Indexing, Primary Indexing, Clustering Indexing, Secondary Indexing over Key, Secondary Indexing over Non Key, Intro to B Tree.

Unit-16: Construction of B Tree, Order of B Tree, Intro to B+ Tree, Construction of B+ Tree, Order of B+ Tree, Min and Max Keys and Nodes in B/B+ Tree, Bulk Loading in B+ Tree, Join Algorithms.

References:

1. Silberschatz, A., Korth, H. F., & Sudarshan, S., *Database System Concepts* (7th ed.). McGraw-Hill Education, 2019
2. Elmasri, R., & Navathe, S. B., *Fundamentals of Database Systems* (7th ed.). Pearson Education, 2015
3. Coronel, C., & Morris, S., *Database Systems: Design, Implementation, & Management* (14th ed.). Cengage Learning, 2022
4. Gillenson, M. L., *Fundamentals of Database Management Systems* (3rd ed.). Wiley, 2023
5. Ramakrishnan, R., & Gehrke, J., *Database Management Systems* (3rd ed.). McGraw-Hill Education, 2003

Course code: ENBO-1401

Credit: 2

Course Title: Basic Knowledge of English Grammar

Course Objectives:

- To develop a fundamental understanding of English grammar rules and concepts.
- To improve students' ability to communicate effectively in both spoken and written English.
- To enhance students' confidence in using correct grammar in various contexts.
- To provide students with the necessary foundation for more advanced studies in English language and literature.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of parts of speech including nouns, pronouns, verbs, adjectives, and adverbs.	L2
CO2	Apply grammatical rules such as subject-verb agreement and correct usage of parts of speech in sentence formation.	L3
CO3	Analyze sentence structures to identify fragments, run-on sentences, and different types of sentences.	L4
CO4	Evaluate and correct sentences based on grammatical accuracy, clarity, and proper structure.	L5
CO5	Construct well-formed sentences by combining ideas effectively using appropriate grammar and sentence structures.	L6

Block I: Parts of Speech

Unit 1: Nouns

Unit 2: Pronouns

Unit 3: Verbs

Unit 4: Adjectives

Unit 5: Adverbs

Block II: Sentence Structure

Unit 6: Subject-Verb Agreement

Unit 7: Sentence Fragments

Unit 8: Run-on Sentences

Unit 9: Types of Sentences

Unit 10: Sentence Combining

References:

1. Murphy, Raymond. *English Grammar in Use: A Self-Study Reference and Practice Book for Intermediate Learners of English*. 5th Edition, Cambridge University Press, 2019.
2. Swan, Michael. *Practical English Usage*. 4th Edition, Oxford University Press, 2016.

3. Barrett, Grant. *Perfect English Grammar: The Indispensable Guide to Excellent Writing and Speaking*. Zephyros Press, 2016.
4. Straus, Jane. *The Blue Book of Grammar and Punctuation: An Easy-to-Use Guide with Clear Rules, Real-World Examples, and Reproducible Quizzes*. 11th Edition, Jossey-Bass, 2014.

Course code: VACO-1401

Credit: 3

Course Title: Understanding India

Course Objectives:

- To provide students with a foundational understanding of the historical events and philosophies that influenced the formation of the Indian Constitution and shaped the nation's identity.
- To familiarize students with the structure, features, and key components of the Indian Constitution, including its unique blend of federalism, parliamentary governance, and judicial independence.
- To examine the fundamental rights and duties outlined in the Constitution, their implications for Indian citizens, and the balance between individual freedoms and social responsibilities.
- To delve into the intricacies of India's system of governance at both the Union and State levels, including the roles and functions of the executive, legislature, and judiciary.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the historical background, basic structure, and salient features of the Indian Constitution, including citizenship and fundamental rights.	L2
CO2	Apply constitutional provisions related to fundamental rights, duties, directive principles, and governance systems in real-life and civic contexts.	L3
CO3	Analyze the structure and functioning of the Indian government, including Parliament, executive bodies, and federal system.	L4
CO4	Evaluate the role and effectiveness of judicial institutions, constitutional and non-constitutional bodies, and mechanisms like judicial review and public interest litigation.	L5
CO5	Develop informed perspectives and solutions regarding governance, rights, and public administration based on constitutional principles.	L6

Block I: The Indian Constitution

Unit 1: Historical Background of the Constitution

Unit 2: Basic structure of the Constitution

Unit 3: Salient Features of Indian Constitution

Unit 4: Union and its Territory, Citizenship

Unit 5: Fundamental Rights

Block II: System of Government

Unit 6: Fundamental Duties & Directive Principles of State Policy

Unit 7: Parliamentary System & Federal system

Unit 8: Parliament

Unit 9: Prime Minister & President
Unit 10: Chief Minister & Governor

Block III: Various Bodies

Unit 11: Panchayati Raj System
Unit 12: Supreme Court & High Court
Unit 13: Judicial Review, Judicial Activism, Public Interest Litigation
Unit 14: Constitutional Bodies
Unit 15: Non - Constitutional Bodies

SUGGESTED READINGS:

Here are some references for Indian polity:

1. Books:

- "Indian Polity" by M. Laxmikanth: A comprehensive guide to Indian constitutional and political systems.
- "Introduction to the Constitution of India" by Durga Das Basu: Provides a detailed analysis of the Indian Constitution and its provisions.
- "Our Constitution: An Introduction to India's Constitution and Constitutional Law" by Subhash C. Kashyap: Offers insights into the principles and functioning of India's constitutional framework.

2. Academic Journals:

- "Economic and Political Weekly": Covers articles and research on various aspects of Indian politics, governance, and public policy.
- "Journal of Indian Law and Society": Focuses on legal and constitutional issues in India, including debates on judicial activism, human rights, and legal reforms.

3. Websites and Online Resources:

- PRS Legislative Research (prsindia.org): Provides analysis, research, and data on Indian parliamentary proceedings, legislation, and policy matters.
- Election Commission of India (eci.gov.in): Offers information on elections, electoral processes, and political parties in India.
- IndiaStat (indiastat.com): Provides statistical data and reports on various aspects of Indian governance, including demographics, economy, and public administration.

4. Government Reports and Documents:

- Reports of the Law Commission of India: Includes recommendations and reports on legal and constitutional reforms in India.
- Annual Reports of the Ministry of Home Affairs and Ministry of Law and Justice: Provide insights into government policies, legislative developments, and law enforcement issues.

These references cover a wide range of topics related to Indian polity, including constitutional law, governance structures, political institutions, electoral processes, and legal frameworks.

Course Code: CABO-1415

Credit: 2

Course Name: Object Oriented Programming with Java Lab

Course Objectives:

- To provide hands-on experience in applying object-oriented programming (OOP) concepts using Java.
- To familiarize students with Java development environments, tools, and libraries.
- To develop proficiency in designing and implementing Java programs using OOP principles such as classes, objects, inheritance, polymorphism, and encapsulation.
- To reinforce theoretical knowledge learned in the Object Oriented Programming with Java course through practical exercises and programming assignments.
- To prepare students to analyze, design, and develop software solutions using Java and OOP paradigms.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain fundamental Java programming concepts including syntax, data types, control structures, object-oriented principles, and exception handling.	L2
CO2	Apply Java programming concepts to develop programs using OOP principles, file handling, GUI development, and database connectivity using JDBC.	L3
CO3	Analyze Java programs involving multithreading, networking, and exception handling to understand program behavior and performance.	L4
CO4	Evaluate Java-based applications based on correctness, efficiency, error handling, and user interaction in GUI and network-based systems.	L5
CO5	Design and develop complete Java applications integrating GUI, database connectivity, multithreading, and networking concepts to solve real-world problems.	L6

Lab Experiments

Lab 1: Write basic Java programs to understand the syntax, data types, operators, and control flow statements in Java.

Lab2: Implement Java classes and objects to understand the concepts of abstraction, encapsulation, inheritance, and polymorphism.

Lab3: Develop graphical user interfaces (GUIs) using Java Swing or JavaFX to create interactive desktop applications.

Lab4: Implement exception handling mechanisms in Java to handle runtime errors and exceptions gracefully.

Lab5: Perform file input/output operations in Java to read from and write to text files, binary files, and streams.

Lab6: Implement multithreaded programs in Java to understand concurrency, synchronization, and thread communication.

Lab7: Connect Java applications to relational databases using JDBC (Java Database Connectivity) for data manipulation and retrieval.

Lab8: Develop networked applications using Java sockets for communication between clients and servers.

Lab9: Design and implement complete Java applications that incorporate multiple concepts learned throughout the course.

References:

1. Farrell, Joyce. *Java Programming*, 9th Edition, Cengage Learning, 2018.
2. Schildt, Herbert. *Java: The Complete Reference*, 11th Edition, McGraw-Hill Education, 2018.
3. Bloch, Joshua. *Effective Java*, 3rd Edition, Addison-Wesley Professional, 2018.
4. Horstmann, Cay S. *Core Java Volume I – Fundamentals*, 11th Edition, Pearson Education, 2018.
5. Deitel, Paul, and Harvey Deitel. *Java: How to Program*, 11th Edition, Pearson Education, 2017.

Semester – V

Course Code: CABO-1501

Credit: 4

Course Name: Software Engineering

Course Objectives:

- To introduce students to the fundamental principles and concepts of software engineering.
- To teach software development methodologies, including requirements analysis, design, implementation, testing, and maintenance.
- To develop skills in software project management, including planning, scheduling, and tracking.
- To explore software quality assurance techniques, such as testing strategies, code reviews, and software metrics.
- To provide hands-on experience with software development tools and techniques through practical projects and CABOe studies.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of software engineering, including software process models, requirements engineering, design principles, and quality assurance.	L2
CO2	Apply software development methodologies, UML modeling techniques, coding standards, and tools like Git to develop and manage software projects.	L3
CO3	Analyze software requirements, system designs, testing strategies, and project workflows using appropriate models and techniques.	L4
CO4	Evaluate software quality, testing methods, project estimation techniques, and risk management strategies based on efficiency, reliability, and industry standards.	L5
CO5	Design and develop software systems using modern engineering practices, including agile methodologies, design patterns, version control, and maintenance strategies.	L6

Block I: Introduction to Software Engineering

Unit 1: Introduction to Software Engineering: Definition and scope of software engineering, Evolution of software engineering methodologies, Importance of software engineering in modern software development.

Unit 2: Software Process Models: Waterfall model, iterative models, incremental model, Agile methodologies: Scrum, Kanban, Extreme Programming (XP), Introduction to DevOps practices and continuous integration/continuous deployment (CI/CD).

Unit 3: Requirements Engineering: Requirements elicitation, analysis, and specification, Functional and non-functional requirements, Use CABOe modeling, user stories, and requirement prioritization.

Block II: Software Design and Implementation

Unit 4: Software Design Principles: Software design process: design principles and patterns, Architectural styles: client-server, layered architecture, microservices,

Unit 5: UML diagrams: class diagrams, sequence diagrams, and state diagrams.

Unit 6: Object-Oriented Design: Object-oriented analysis and design (OOAD) concepts, Design patterns: creational, structural, and behavioral patterns, Application of design patterns in software development.

Unit 7: Implementation and Coding Standards: Coding standards and guidelines, Best practices in coding: modularity, readability, and maintainability, Introduction to version control systems (e.g., Git) and collaborative development.

Block III: Software Testing and Quality Assurance

Unit 8: Software Testing Fundamentals: Testing levels: unit testing, integration testing, system testing, Testing techniques: black-box testing, white-box testing, and grey-box testing, Test CABOe design, test coverage,

Unit 9: Test automation: Introduction to test automation tools (JUnit, Selenium), Test-driven development (TDD), Continuous integration and continuous testing

Unit 10: Software Quality Assurance: Quality assurance processes and techniques, Software metrics: measuring software quality and productivity, Quality standards and certifications: ISO 9001, CMMI, and IEEE standards.

Unit 11: Project Planning and Estimation: Project management processes: planning, scheduling, and resource allocation,

Block IV: Software Project Management

Unit 12: Project Estimation techniques: effort estimation, cost estimation, and time estimation, Project management tools and techniques: Gantt charts, PERT charts, and agile planning boards.

Unit 13: Risk Management: Risk identification, analysis, and mitigation strategies, Risk management in software projects: proactive and reactive approaches, Contingency planning and risk monitoring throughout the project lifecycle.

Unit 14: Agile Project Management: Agile project management principles and practices, Scrum framework: roles, ceremonies, and artifacts, Agile metrics and performance measurement in agile projects.

Unit 15: Software Maintenance and Evolution: Software maintenance activities: corrective, adaptive, and perfective maintenance, Software reengineering and refactoring techniques,

Unit 16: Software Evolution: Managing software evolution: versioning, release management, and software configuration management.

References:

1. Pressman, Roger S. "Software Engineering: A Practitioner's Approach." McGraw-Hill Education, 2024.
2. Sommerville, Ian. "Software Engineering." Pearson, 2015.
3. Sommerville, Ian. "Software Engineering: International Edition." Pearson, 2021.
4. Pfleeger, Shari Lawrence, and Joanne M. Atlee. "Software Engineering: Theory and Practice." Pearson, 2018.
5. Shari Lawrence Pfleeger and Joanne M. Atlee, *Software Engineering: Theory and Practice*, 4th ed., Pearson, 2010.
6. McConnell, Steve. "Code Complete: A Practical Handbook of Software Construction." Microsoft Press, 2004.

Course Code: CABO-1502

Credit: 4

Course Name: Computer Networks

Course Objectives:

- To provide a comprehensive understanding of computer networks and their components.
- To teach networking concepts, protocols, and architectures, including OSI and TCP/IP models.
- To develop proficiency in configuring and troubleshooting network devices such as routers, switches, and firewalls.
- To explore advanced topics such as network security, wireless networking, and network management.
- To prepare students for industry certifications such as Cisco Certified Network Associate (CCNA) by covering relevant topics and skills.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of data communication, transmission techniques, network topologies, transmission media, and network models such as OSI and TCP/IP.	L2
CO2	Apply data transmission, error detection and correction methods, and networking protocols (TCP, UDP, IPv4/IPv6) to solve communication and networking problems.	L3
CO3	Analyze network architectures, routing algorithms, flow control mechanisms, and MAC protocols for efficient data communication.	L4
CO4	Evaluate network performance, addressing schemes, subnetting techniques, and security mechanisms based on efficiency, reliability, and scalability.	L5
CO5	Design and develop network solutions incorporating routing, IP addressing, transport protocols, and security techniques for real-world communication systems.	L6

Block 1: Fundamentals of Data Communication

Unit 1: Data Communication: Components of a Data Communication System, Simplex, Half- Duplex and Duplex Modes of Communication; Analog and Digital Signals; Noiseless and Noisy Channels; Bandwidth, Throughput and Latency;

Unit 2: Digital and Analog Transmission: Digital and Analog Transmission; Data Encoding and Modulation Techniques; Broadband and Baseband Transmission; Multiplexing, Transmission Media, Transmission Errors, Error Handling Mechanisms.

Unit 3: Computer Networks: Network Topologies, Local Area Networks, Metropolitan Area Networks, Wide Area Network, Wireless Networks, Internet.

Unit 4: Transmission Media, Guided & Unguided Transmission Media: Twisted pair, coaxial cable, fiber-optic cable, Characteristics, advantages, and disadvantages of each type of guided media, Wireless Transmission, Radio waves, microwaves, infrared, Wireless communication technologies: Wi-Fi, Bluetooth, cellular networks

Block – II: Computer Network and Reference Models

Unit 5: Network Models: Layered Architecture, OSI Reference Model and its Protocols; TCP/IP Protocol Suite, Physical, Logical, Port and Specific Addresses; Switching Techniques.

Unit 6: Functions of OSI and TCP/IP Layers: Use, Purpose, Origin, Functionality, Usability etc.

Unit 7: Framing, Error Detection and Correction; Flow and Error Control; Sliding Window Protocol, HDLC, Multiple Access – CSMA/CD, CSMA/CA, Reservation, Polling, Token Passing,

Unit 8: IPv4 Structure and Address: IPv4 Structure and Address Space; Classful and Classless Addressing; Datagram, Fragmentation and Checksum; IPv6 Packet Format, Mapping Logical to Physical Address (ARP), Direct and Indirect Network Layer Delivery;

Block – III: Network Protocols and Layers

Unit 9: Routing Algorithms & Protocols: TCP, UDP and SCTP Protocols; Flow Control, Error Control and Congestion Control in TCP and SCTP.

Unit 10: Data Link Layer Error Detection and Correction: Error detection methods: parity check, checksum, cyclic redundancy check (CRC), Error correction techniques: Hamming codes, Reed-Solomon codes

Unit 11: Medium Access Control (MAC): MAC protocols: CSMA/CD, CSMA/CA, token passing, Ethernet and IEEE 802 standards for LANs

Unit 12: Network Layer: Routing Algorithms - Shortest path algorithms: Dijkstra's algorithm, Bellman-Ford algorithm - Distance vector and link-state routing protocols

Block – IV: Advanced Network Concepts

Unit 13: IP Addressing and Subnetting: IPv4 and IPv6 addressing schemes, Subnetting and subnet mask calculation

Unit 14: Transport Layer: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) - Characteristics and differences between TCP and UDP - Reliability, flow control, and congestion control in TCP, TCP/IP Applications, Common TCP/IP applications: HTTP, FTP, SMTP, DNS, Overview of application layer protocols and their functions

Unit 15: World Wide Web (WWW): Uniform Resource Locator (URL), Domain Name Service (DNS), Resolution - Mapping Names to Addresses and Addresses to Names; Electronic Mail Architecture, SMTP, POP and IMAP; TELNET and FTP.

Unit 16: Network Security: Malwares, Cryptography and Steganography; Secret-Key Algorithms, Public-Key Algorithms, Digital Signature, Virtual Private Networks, Firewalls.

References:

1. Behrouz A Forouzan, Data Communication and Computer networks, 4th ed, McGraw Hill. (2022)
2. A.S. Tanenbaum, “Computer Networks”, 4th Edition, Prentice Hall India, (2022)
3. S. Keshav, “An Engineering Approach on Computer Networking”, Addison Wesley, 2012
4. W. Stallings, “Data and Computer Communication”, Macmillan Press, 2021.
5. James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, 8th ed., Pearson, 2020

Course Code: CABO-1503

Credit: 4

Course Name: Theory of Computation

Course Objectives:

- To introduce students to the theoretical foundations of computation and formal languages.
- To teach key concepts such as automata theory, formal grammars, and Turing machines.
- To develop proficiency in analyzing the computational power and limitations of different models of computation.
- To explore topics such as regular languages, context-free languages, and computability theory.
- To provide insight into the relationships between theoretical models of computation and practical applications in computer science.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of automata theory, formal languages, grammars, and models of computation including finite automata, pushdown automata, and Turing machines.	L2
CO2	Apply formal methods such as regular expressions, finite automata, context-free grammars, and parsing techniques to solve language recognition problems.	L3
CO3	Analyze computational models, language classes, and problems using pumping lemma, closure properties, and equivalence between automata and grammars.	L4
CO4	Evaluate computability and complexity of problems using concepts like decidability, Turing machines, halting problem, and Chomsky hierarchy.	L5
CO5	Design and develop computational models and state machines for real-world applications, including software modeling and language processing systems.	L6

Block I: Introduction to Automata Theory

Unit 1: Introduction to Theory of Computation, Overview of computational models and their applications, Introduction to automata theory and its importance in computer science

Unit 2: Finite Automata and Regular Languages, Definition and properties of finite automata, Regular languages and regular expressions, Finite automata and regular expression equivalence

Unit 3: REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA.

Unit 4: Pumping Lemma & Closure Properties: Proving languages to be non-regular -Pumping lemma, applications, Closureproperties of regular languages

Unit 5: Context-Free Grammars and Languages, Context-free grammars and their derivations, Context-free languages and their properties, Parsing techniques: top-down and bottom-up parsing

Unit 6: PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA.

Block II: Turing Machines and Computability

Unit 7: Turing Machines, Definition and components of Turing machines, Turing machine variants: multi-tape, non-deterministic, Universal Turing machine and its significance

Unit 8: Computability and Undecidability, Church-Turing thesis and computability, Halting problem and its undecidability,

Unit 9: Introduction to recursive and recursively enumerable languages, Context sensitive language and linear bounded automata (LBA),

Unit 10: Undecidable problems about TMs. Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

Block III: Complexity Theory

Unit 11: Formal Language Processing, Introduction to formal language processing,

Unit 12: Regular expressions and their applications, Applications of context-free grammars in parsing

Unit 13: Finite State Machines in Software Development, State machines in software modeling,

Unit 14: Using finite state machines for program logic, Finite state machine-based design patterns

Block IV: Finite State Machines in Software Development

Unit 15: Finite State Machines, State machines in software modeling, Using finite state machines for program logic, Finite state machine-based design patterns

Unit 16: Formal Language Processing, Introduction to formal language processing, Regular expressions and their applications, Applications of context-free grammars in parsing

References:

1. "Introduction to Automata Theory, Languages, and Computation" by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman (2006)
2. K. L. P. Mishra, N. Chandra shekaran (2003), Theory of Computer Science- Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.(2019)
3. Automata and Computability"Dexter C. Kozen (2022)
4. "Elements of the Theory of Computation"Harry R. Lewis, Christos H. Papadimitriou (1997, Pearson)
5. "Introduction to the Theory of Computation"Michael Sipser 3rd Edition (2012, Cengage Learning)

Course Code: CABO – 1504

Credit: 3

Course Name: Perspectives in Physics

Course Objectives:

- To introduce students to the foundational principles and theories of physics.
- To explore the historical development of physics and its role in understanding the natural world.
- To provide an overview of key concepts in classical mechanics, electromagnetism, thermodynamics, and modern physics.
- To foster critical thinking skills by examining the philosophical implications and interdisciplinary connections of physics.
- To cultivate an appreciation for the beauty and complexity of the physical universe through engaging discussions and experiments.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of atomic structure, quantum mechanics, nuclear physics, relativity, and elementary particles.	L2
CO2	Apply principles of atomic models, radioactive decay, nuclear reactions, and relativistic effects to solve numerical and conceptual problems.	L3
CO3	Analyze physical phenomena related to particle waves, nuclear energy, and relativistic motion using appropriate theoretical models.	L4
CO4	Evaluate the implications of nuclear energy, relativity, and particle physics concepts in real-world applications and scientific advancements.	L5
CO5	Design and interpret theoretical models or simple experiments to demonstrate concepts such as quantum behavior, nuclear processes, and cosmological theories.	L6

Block-I: The Atom and particle waves

Unit-1: The structure of the atom, The existence of atoms: Evidence from chemistry,

Unit-2: Cathode rays, Electrons, and X rays, Atomic spectra and the Bohr model of the atom,

Unit-3: Particle waves and Quantum Mechanics.

Block-II: The Nucleus and Nuclear Energy

Unit-4: Radioactivity and the discovery of the nucleus, The structure of the nucleus, Radioactive decay,

Unit-5: Nuclear reactions and nuclear fission, nuclear reactors,

Unit-6: Nuclear weapons and nuclear fusion

Block-III: Special Relativity and Beyond

Unit-7: Relative motion in classical physics. The speed of light and Einstein's postulates.

Unit-8: Time dilation and length contraction.

Unit-9: Newton's laws and mass-energy equivalence. General relativity (Idea).

Block-IV: Idea of Elementary Particles, Cosmology

Unit-10: Quarks and other elementary particles.

Unit-11: Cosmology and the beginning of time

References:

1. The Physics of Everyday Phenomena: A Conceptual Introduction ToPhysics, Sixth Edition
W. Thomas Griffith & Juliet W. BrosingMcgraw-Hill,(2024)
2. IntroductionToPhysicsJohnDCutnell.K.W.Johnson,DYoung,SStadlerWileyIndiaEdition,
(2022)
3. Fundamentals Of Physics D Halliday,R Resnick, J Walker John Wiley And Sons,(2021)
4. For The Love Of Physics Walter Levin Free Press,(2011)
5. Concepts Of Modern Physics A Beiser, S Mahajan, Mc Graw Hill,(2023).

Course Code: CABO-1505

Credit: 3

Course Name: Basic Abstract & Linear Algebra

Course Objectives:

- To provide a foundational understanding of abstract algebraic structures such as groups, rings, and fields.
- To introduce students to linear algebra concepts including vector spaces, matrices, and linear transformations.
- To teach fundamental operations and properties of algebraic structures and matrices.
- To develop proficiency in solving linear systems of equations and performing matrix operations.
- To explore applications of abstract and linear algebra in various fields such as physics, engineering, and computer science.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of abstract algebra, including groups, subgroups, cyclic groups, permutation groups, rings, and fields.	L2
CO2	Apply concepts of vector spaces, linear combinations, basis, and dimension to solve algebraic problems.	L3
CO3	Analyze algebraic structures such as cosets, quotient spaces, and linear transformations to understand their properties and relationships.	L4
CO4	Evaluate properties of algebraic systems using theorems such as Lagrange's theorem and determine invertibility and equivalence of transformations.	L5
CO5	Design and represent algebraic models using matrices and linear transformations to solve advanced mathematical and computational problems.	L6

Block I: Basic Abstract Algebra

Unit 1: Group & Sub Groups

Unit 2: Cyclic Group & Permutation Group

Unit 3: Isomorphic, Coset & Lagrange's Theorem

Unit 4: Ring & Field

Block II: Theory of Vector Spaces

Unit 5: Vector spaces: Definition and Examples

Unit 6: Linear combination, linear span, linear independence and dependence

Unit 7: Basis and Dimension of a vector space

Unit 8: Quotient space and Direct sum of vector spaces

Block III: Theory of Linear Transformation and Linear Operators

Unit 9: Linear Transformations

Unit 10: Space of Linear Transformation

Unit 11: Invertible Linear Transformation

Unit 12: Matrix representation of Linear Transformation

References:

1. Gallian, Joseph. A. (2021). Contemporary Abstract Algebra (8thed.).
2. Cengage Learning India Private Limited, Delhi. Fourth impression,2015.
3. Cheney,Ward,andDavidKincaid."Linearalgebra:Theoryandapplications."*The Australian Mathematical Society* 110 (2009):544-550.
4. Axler, Sheldon. *Linear algebra done right*. Springer Nature,2024.
5. Spence,LawrenceE.,ArnoldJ.Insel,andStephenH.Friedberg.*Elementarylinear algebra*.2000.
6. Khan, Mohd. *Lectures on linear algebra*. Anamaya Publishers(2008).

Course Code: CABO-1516

Credit: 2

Course Name: Software Engineering Lab

Course Objectives:

- To offer practical experience in applying software engineering principles and methodologies to real-world projects.
- To familiarize students with software development tools, techniques, and best practices.
- To provide hands-on experience in requirements gathering, software design, implementation, testing, and maintenance.
- To foster teamwork and collaboration skills through group projects and team-based activities.
- To prepare students for the challenges and responsibilities of software engineering roles in industry by simulating real-world project scenarios and environments.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain key software engineering concepts including SRS documentation, UML diagrams, version control, testing, and agile practices.	L2
CO2	Apply tools and techniques such as UML modeling, version control using Git, and testing frameworks like JUnit or pytest to develop and manage software projects.	L3
CO3	Analyze software requirements, system workflows, and design structures using diagrams, test cases, and design patterns.	L4
CO4	Evaluate software quality through code reviews, testing strategies, effort estimation techniques, and the effectiveness of design patterns.	L5
CO5	Design and implement software solutions incorporating proper documentation, version control practices, testing frameworks, and design patterns for real-world applications.	L6

Lab Experiment Questions

Lab 1. What is the primary purpose of the SRS document?

Lab 2. How do you distinguish between functional and non-functional requirements in your project?

Lab 3. What is the significance of class diagrams in system design?

Lab 4. Create a sequence diagram for a key use CABOe in your project. How does it help in understanding the system's workflow?

Lab 5. What are the primary advantages of using version control systems like Git in software development?

Lab 6. Demonstrate how to create and switch between branches in Git. Why is branching important?

Lab 7. Explain the process of committing changes to a Git repository. How do commit messages improve collaboration?

Lab 8. How do you estimate effort using story points during sprint planning? Provide an example.

Lab 9. Describe the format and significance of a daily stand-up meeting.

- Lab 10.** What are the key components of a test CABOe?
- Lab 11.** Write test CABOes for a specific functionality in your project, including input, execution conditions, and expected results.
- Lab 12.** How do you use a testing framework (e.g., JUnit, pytest) to automate test execution? Provide an example.
- Lab 13.** Choose a design pattern (e.g., Singleton, Factory, Observer) and explain its purpose.
- Lab 14.** Implement the chosen design pattern in your project and demonstrate its working with sample scenarios.
- Lab 15.** What are the benefits and potential drawbacks of using this design pattern?
- Lab 16.** What are the key aspects to look for during a code review?

References:

1. Sommerville, Ian. (2022). *Software Engineering* (10th ed.). Pearson
2. Pressman, Roger S. (2020). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
3. Martin, Robert C. (2020). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.
4. Beck, Kent. (2018). *Extreme Programming Explained: Embrace Change* (2nd ed.). Addison-Wesley.
5. Fowler, Martin. (2021). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

Semester – VI

Course Code: CABO–1601

Credit: 4

Course Name: IoT & Cloud Computing

Course Objectives:

- To introduce students to the concepts and technologies of the Internet of Things (IoT) and cloud computing.
- To teach the architecture and components of IoT systems and cloud platforms.
- To explore protocols, standards, and communication technologies used in IoT and cloud environments.
- To provide hands-on experience in developing and deploying IoT applications on cloud platforms.
- To investigate security, privacy, and scalability challenges in IoT and cloud computing, along with strategies for addressing them.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of IoT and cloud computing, including architectures, protocols, service models, and their integration.	L2
CO2	Apply IoT device programming, communication protocols, and cloud services (such as AWS, Microsoft Azure) to develop and deploy IoT applications.	L3
CO3	Analyze IoT systems and cloud infrastructures in terms of scalability, data communication, security challenges, and performance requirements.	L4
CO4	Evaluate different IoT protocols, cloud deployment models, and data analytics frameworks based on efficiency, security, and suitability for real-world applications.	L5
CO5	Design and develop end-to-end IoT solutions integrating devices, cloud platforms, data processing, visualization, and security mechanisms.	L6

Block I: Introduction to IoT and Cloud Computing

Unit 1: Introduction to IoT: Definition and concepts of IoT, Evolution and significance of IoT, IoT architecture and components

Unit 2: Introduction to Cloud Computing: Definition and characteristics of cloud computing, Cloud service models: IaaS, PaaS, SaaS, Cloud deployment models: Public, private, hybrid, community clouds

Unit 3: Convergence of IoT and Cloud Computing: Role of cloud computing in IoT ecosystem, Benefits and challenges of integrating IoT with cloud computing, CABOe studies showCABOing IoT and cloud computing synergy

Unit 4: IoT Protocols and Standards: Communication protocols: MQTT, CoAP, HTTP, etc., IoT standards: IEEE 802.15.4, Zigbee, LoRaWAN, etc., Security considerations in IoT protocols

Block II: IoT Device Development

Unit 5: IoT Hardware Platforms: Overview of popular IoT hardware platforms, Selection criteria for IoT hardware, Hands-on exercises with IoT development boards

Unit 6: IoT Device Programming: Introduction to IoT programming languages: Python, C, JavaScript, Basics of sensor interfacing and data acquisition, Hands-on IoT programming exercises

Unit 7: IoT Device Communication: Wireless communication technologies for IoT devices: Wi-Fi, Bluetooth, RFID, etc., Setting up communication between IoT devices and cloud servers, Data transmission protocols and techniques

Unit 8: IoT Device Security: Security challenges in IoT devices, Secure coding practices for IoT devices, Encryption and authentication mechanisms for IoT communication

Block III: Cloud Infrastructure and Services

Unit 9: Cloud Infrastructure Basics, Virtualization technology and concepts, Cloud storage solutions: Object storage, block storage, file storage, Introduction to cloud computing platforms: AWS, Azure, Google Cloud, etc.

Unit 10: Cloud Service Deployment: Deploying applications and services on cloud platforms Containerization technologies: Docker, Kubernetes, Hands-on exercises on deploying IoT applications on the cloud

Unit 11: Scalability and Elasticity in Cloud Computing: Horizontal and vertical scaling Auto-scaling and load balancing techniques, Ensuring scalability for IoT applications on the cloud

Unit 12: Cloud Security: Cloud security challenges and threats, Identity and access management (IAM), Encryption and data protection in the cloud

Block IV: IoT Data Management and Analytics

Unit 13: Data Collection and Storage for IoT: Data ingestion mechanisms for IoT applications Time-series databases for IoT data storage, Data retention and archival strategies

Unit 14: IoT Data Processing and Analysis: Real-time data processing techniques: Stream processing, Complex event processing (CEP), Data analytics frameworks for IoT: Apache Kafka, Apache Spark, etc., Extracting insights from IoT data

Unit 15: IoT Data Visualization: Data visualization techniques for IoT data, Dashboard design and development for IoT applications, Tools and libraries for IoT data visualization

Unit 16: IoT Data Security and Privacy: Securing IoT data at rest and in transit, Privacy concerns in IoT data collection and processing, Compliance with data protection regulations (e.g., GDPR)

References:

1. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes, Gonzalo Salgueiro, Patrick Grossetete(2024)
2. "Edge Computing: A Primer" by AgustinusBorgyWaluyo (2021)
3. "Fog and Edge Computing: Principles and Paradigms" edited by Rajkumar Buyya, Satish Narayana Srirama, and Shadi Ibrahim (2019)
4. "Edge Computing: From Hype to Reality" Fawzi Behmann, Kwok Wust Edition (2023, Wiley)
5. "Building the Internet of Things with IPv6 and MQTT" Chandrasekar Vuppalapati 1st Edition (2023, Apress)

Course Code: CABO–1602

Credit: 4

Course Name: Machine Learning using Python

Course Objectives:

- To introduce students to the fundamental concepts and algorithms of machine learning.
- To teach the Python programming language and its libraries for machine learning such as NumPy, Pandas, and Scikit-learn.
- To explore supervised and unsupervised learning techniques including regression, classification, clustering, and dimensionality reduction.
- To provide hands-on experience in implementing machine learning algorithms and models using Python.
- To foster critical thinking and problem-solving skills through practical exercises and real-world applications of machine learning.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of machine learning, Python programming basics, data preprocessing techniques, and types of learning (supervised, unsupervised).	L2
CO2	Apply Python-based tools and libraries such as NumPy, Pandas, and Scikit-learn to preprocess data and implement machine learning models.	L3
CO3	Analyze datasets using supervised and unsupervised learning algorithms such as regression, decision trees, SVM, clustering, and dimensionality reduction techniques.	L4
CO4	Evaluate the performance of machine learning models using appropriate metrics and techniques, including model validation, hyperparameter tuning, and comparison of algorithms.	L5
CO5	Design and develop intelligent systems using advanced techniques such as neural networks, CNNs, RNNs, and transfer learning for real-world applications.	L6

Block I: Introduction to Machine Learning and Python Basics

Unit 1: Introduction to Machine Learning: Overview of machine learning concepts, types, and applications, Understanding supervised, unsupervised, and reinforcement learning, Introduction to the Python ecosystem for machine learning: NumPy, Pandas, Matplotlib, and Scikit-learn

Unit 2: Python Basics for Machine Learning: Introduction to Python programming language: syntax, data types, variables, Control structures: loops and conditional statements, Functions and modules: defining functions, importing modules

Unit 3: Data Preprocessing in Python: Data cleaning and preprocessing techniques: handling missing values, outliers, and duplicates, Data exploration and visualization using Matplotlib and Seaborn, Feature scaling and normalization for improving model performance

Unit 4: Introduction to Supervised Learning: Overview of supervised learning: classification and regression tasks, Understanding the concepts of features, labels, and training data, Introduction to popular supervised learning algorithms: Linear Regression, Logistic Regression

Block II: Supervised Learning Algorithms

Unit 5: Linear Regression: Understanding linear regression: assumptions, cost function, gradient descent, Implementing linear regression in Python using Scikit-learn, Evaluating model performance: metrics like mean squared error, R-squared

Unit 6: Logistic Regression: Introduction to logistic regression for binary classification, Logistic regression cost function, sigmoid function, and decision boundary, Implementing logistic regression in Python and evaluating model performance

Unit 7: Decision Trees and Random Forests: Introduction to decision trees: construction, splitting criteria, pruning, Ensemble learning with Random Forests: bagging and boosting, Implementation and evaluation of decision trees and random forests in Python

Unit 8: Support Vector Machines (SVM): Understanding Support Vector Machines: margin, kernels, hyperplanes, Implementing SVM for classification tasks using Scikit-learn, Tuning SVM hyperparameters and evaluating model performance

Block III: Unsupervised Learning and Dimensionality Reduction

Unit 9: Introduction to Unsupervised Learning: Overview of unsupervised learning: clustering and dimensionality reduction, Introduction to K-means clustering algorithm and its implementation in Python, Evaluating clustering performance using metrics like silhouette score

Unit 10: Hierarchical Clustering and Density-Based Clustering: Introduction to hierarchical clustering: agglomerative and divisive clustering, Density-based clustering with DBSCAN algorithm, Comparing different clustering algorithms and selecting appropriate ones for different datasets

Unit 11: Principal Component Analysis (PCA): Understanding dimensionality reduction with PCA: covariance matrix, eigenvectors, eigenvalues, Implementing PCA for feature extraction and visualization in Python, Interpreting PCA results and selecting the number of principal components

Unit 12: t-Distributed Stochastic Neighbor Embedding (t-SNE), Introduction to t-SNE for visualization of high-dimensional data, Implementing t-SNE in Python and visualizing high-dimensional datasets, Understanding the limitations and considerations when using t-SNE

Block IV: Advanced Topics in Machine Learning

Unit 13: Introduction to Neural Networks: Overview of artificial neural networks (ANNs) and deep learning,

Unit 14: Basics of feedforward neural networks: layers, activation functions, forward propagation, Implementing a simple neural network for classification tasks in Python using TensorFlow/Keras

Unit 15: Convolutional Neural Networks (CNNs): Introduction to CNNs, convolutional layers, pooling layers, fully connected layers, Implementing CNNs for image classification tasks using TensorFlow/Keras,

Unit 16: Transfer learning with pre-trained CNN models like VGG, ResNet, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM): Introduction to RNNs and LSTM networks for sequential data processing, Implementing RNNs and LSTM networks for sequence prediction tasks in Python, Applications of RNNs and LSTM in natural language processing (NLP) and time series analysis

References:

1. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Geron (2022)
2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili (2019)
3. "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido (2016)
4. "Deep Learning with Python" by François Chollet (2021)

Course Code: CABO–1603

Credit: 4

Course Name: Discrete Mathematics

Course Objectives:

- To provide a foundational understanding of key concepts in discrete mathematics, including sets, functions, relations, and logic.
- To introduce students to fundamental topics such as combinatorics, graph theory, and discrete probability.
- To teach methods for analyzing and solving problems using discrete mathematical techniques.
- To explore applications of discrete mathematics in computer science, cryptography, and other fields.
- To develop critical thinking and problem-solving skills through rigorous mathematical reasoning and proof techniques.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of discrete mathematics, including set theory, propositional and predicate logic, relations, Boolean algebra, and graph theory.	L2
CO2	Apply combinatorial techniques such as permutations, combinations, binomial theorem, and counting principles to solve mathematical problems.	L3
CO3	Analyze discrete structures including relations, graphs, and logical expressions using appropriate algorithms and mathematical reasoning.	L4
CO4	Evaluate probabilistic models, number theory concepts, and cryptographic techniques based on correctness, efficiency, and applicability.	L5
CO5	Design and develop solutions to computational problems using discrete mathematics concepts such as graph algorithms, logical reasoning, and cryptographic methods.	L6

Block I: Foundations of Discrete Mathematics

Unit 1: Introduction to Discrete Mathematics, Basic terminologies and concepts

Unit 2: Set Theory and Operations: Sets, operations on sets, Venn diagrams, set identities

Unit 3: Propositional Logic and Truth Tables: Propositions, truth tables, logical connectives, logical equivalence, laws of logic.

Unit 4: Predicate Logic and Quantifiers: Predicates, quantifiers, logical inference, universal and existential quantification.

Block II: Combinatorics and Counting Techniques

Unit 5: Permutations and Combinations: Fundamental principles, permutations with repetitions, combinations with repetitions.

Unit 6: Binomial Theorem and Applications: Binomial coefficients, Pascal's triangle, binomial theorem.

Unit 7: Principle of Inclusion-Exclusion: Counting techniques for overlapping sets.

Unit 8: Pigeonhole Principle and Applications: Applications of the pigeonhole principle in combinatorial problems.

Block III: Graph Theory and Discrete Structures

Unit 9: Introduction to Discrete Structures: Relations and Partial Orders: Relations, equivalence relations, partial orders, lattices.

Unit 10: Boolean Algebra

Unit 11: Introduction to Graph Theory: Basic definitions, types of graphs, graph representations.

Unit 12: Graph Traversals and Algorithms: Depth-first search (DFS) and Breadth-first search (BFS).

Block III: Discrete Probability and Number Theory

Unit 13: Basics of Discrete Probability: Sample spaces, events, probability axioms, conditional probability.

Unit 14: Random Variables and Distributions: Discrete random variables, probability mass functions, expectation and variance.

Unit 15: Introduction to Number Theory: Divisibility, prime numbers, modular arithmetic.

Unit 16: Cryptography and Applications: Basics of cryptography, encryption algorithms based on number theory principles.

References:

1. Discrete Mathematical Structures, Kolman, Busby & Ross: PHI, 6th Edition, 2009.
2. Elements of Discrete Maths, C.L. Liu : Tata McGraw Hill, 2nd edition, 2001.
3. Discrete Mathematics and Its Applications: By Kenneth H Rosen, McGraw Hill, 8th edition 2019.
4. J. P. Tremblay, R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill Pub, 1975.

Course code: CABO-1604

Credit: 3

Course Name: Concepts of Physics

Course Objectives:

- To introduce students to the fundamental principles and laws of physics.
- To teach key concepts such as motion, forces, energy, and momentum.
- To develop an understanding of classical mechanics and its applications in describing the motion of objects.
- To explore basic principles of electricity and magnetism.
- To foster critical thinking skills through problem-solving exercises and hands-on experiments to illustrate physics concepts.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of physics, including motion, energy, electricity, magnetism, waves, optics, fluids, and thermodynamics.	L2
CO2	Apply principles such as Newton's laws, Coulomb's law, and laws of thermodynamics to solve numerical and real-world physics problems.	L3
CO3	Analyze physical phenomena involving motion, electromagnetic effects, wave behavior, and fluid dynamics using appropriate laws and mathematical models.	L4
CO4	Evaluate different physical systems and processes such as heat engines, optical instruments, and electrical circuits based on efficiency, behavior, and performance.	L5
CO5	Design and interpret simple experiments, models, or simulations to demonstrate key physics principles and their real-life applications.	L6

Block-I: Physics, the Fundamental Science, and Motion

Unit-1: The scope of Physics, The role of measurement and mathematics in Physics, Physics and everyday phenomena and home experiments and observations.

Unit-2: Describing motion, falling objects and projectile motion, Newton's laws: Explaining motion, Circular motion, the planets, and gravity,

Unit-3: Energy and oscillations, Momentum and impulse, Rotational motion of solid objects.

Block-II: Electricity and Magnetism

Unit-4: Electrostatic phenomena: The electrostatic force: Coulomb's law, The electric field and electric potential, Conductors and insulators, Electric circuits

Unit-5: Magneto-statics: Magnets, magnetic force and magnetic effects of electric currents,

Unit-6: Faraday's law: Electromagnetic induction.

Block-III: Wave Motion and Optics

Unit-7: Waves: Wave pulses and periodic wave, Interference and standing waves, Sound waves and the physics of music.

Unit-8: Light waves and colour: Electromagnetic waves, wavelength and colour, Interference of light waves,

Unit-9: Diffraction, gratings, polarized light, Light and image formation, Eyeglasses, Microscopes, Telescopes.

Block-IV: Fluids and Heat

Unit-10: The behavior of fluids, Pressure and Pascal's principle, Archimedes' principle, Fluids in motion,

Unit-11: Bernoulli's principle, Temperature, and heat,

Unit-12: Heat engines and the law of thermodynamics. Concept of entropy.

References:

1. The Physics Of Everyday Phenomena: A Conceptual Introduction To Physics, Sixth Edition W. Thomas Griffith, Juliet W. Brosing McGraw-Hill, 9th edition (2019)
2. Introduction to Physics John D Cutnell. K. W. Johnson, D Young, S Stadler Wiley India Edition, (2022)
3. Fundamentals of Physics D Halliday, R Resnick, J Walker John Wiley And Sons, 5th edition (2021)
4. For the Love of Physics Walter Levin Free Press (2011) The Feynman
5. Lectures On Physics Vol. I R Feynman, Leighton, Sands Narosa Publishing House, (2008)

Course Code: CABO–1615

Credit: 2

Course Name: Machine Learning Using Python Lab

Course Objectives:

- To provide practical experience in implementing machine learning algorithms and techniques using Python.
- To familiarize students with popular machine learning libraries and frameworks such as Scikit-learn, TensorFlow, and Keras.
- To offer hands-on exercises and projects to reinforce theoretical concepts learned in machine learning courses.
- To develop skills in data preprocessing, model training, evaluation, and deployment in Python.
- To prepare students to apply machine learning methods to real-world datasets and problems through guided lab sessions and projects.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the basic concepts of data analysis, data preprocessing, visualization, and machine learning techniques used in practical datasets.	L2
CO2	Apply data handling, preprocessing, and visualization techniques using tools like Pandas and Keras to prepare datasets for analysis.	L3
CO3	Analyze datasets using statistical measures, clustering methods, dimensionality reduction, and classification algorithms such as logistic regression and decision trees.	L4
CO4	Evaluate the performance of machine learning and text classification models using metrics like accuracy, precision, recall, and interpret results effectively.	L5
CO5	Design and develop machine learning solutions, including classification models, clustering approaches, neural networks, and text processing systems for real-world datasets.	L6

Lab Experiment

- Lab-1. Load a dataset using Pandas and display the first few rows to understand its structure.
- Lab-2. Explore basic statistics such as mean, median, and standard deviation for numerical columns.
- Lab-3. Visualize data distributions using histograms and box plots.
- Lab-4. Handle missing values in the dataset using methods like mean imputation or dropping rows/columns.
- Lab-5. Encode categorical variables using techniques like one-hot encoding or label encoding.
- Lab-6. Scale numerical features using techniques like Min-Max scaling or standardization.
- Lab-7. Implement logistic regression to classify data into two classes based on input features.
- Lab-8. Build a decision tree classifier to predict the class labels of samples based on input features.
- Lab-9. Visualize the decision tree structure and interpret the decision rules.
- Lab-10. Determine the optimal number of clusters using the elbow method or silhouette score.

- Lab-11. Understand the explained variance ratio of principal components.
- Lab-12. Build a simple feedforward neural network using the Keras library to classify images from the MNIST dataset.
- Lab-13. Preprocess text data by tokenizing, stemming, and vectorizing text documents.
- Lab-14. Build a text classification model using techniques such as bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), and word embeddings.
- Lab-15. Evaluate the performance of the text classification model using metrics like accuracy, precision, and recall.

References:

1. Géron, Aurélien. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow." O'Reilly Media, 2019.
2. Raschka, Sebastian, and Mirjalili, Vahid. "Python Machine Learning." Packt Publishing, 2017.
3. Müller, Andreas C., and Guido, Sarah. "Introduction to Machine Learning with Python: A Guide for Data Scientists." O'Reilly Media, 2016.
4. Chollet, François. "Deep Learning with Python." Manning Publications, 2018.
5. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" Géron, Aurélien. 3rd Edition (2022)

Course Code: CABO–1691

Credit: 3

Course Name: Internship

Course Objectives:

- To provide students with practical exposure to the industry and real-world work environments.
- To allow students to apply theoretical knowledge gained during their BCA program to practical projects and tasks.
- To develop professional skills such as teamwork, communication, problem-solving, and time management.
- To enable students to gain hands-on experience in specific areas of interest within the field of computer science, such as software development, web development, database management, networking, or cybersecurity.
- To facilitate networking opportunities and professional connections that can potentially lead to job placements or future career opportunities after completing the internship.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the organizational structure, work environment, and professional practices followed in the industry or organization during the internship.	L2
CO2	Apply technical knowledge, tools, and skills acquired during the academic program to perform assigned tasks and solve real-world problems in the workplace.	L3
CO3	Analyze workplace problems, project requirements, and workflows to identify appropriate solutions and improve efficiency.	L4
CO4	Evaluate the effectiveness of implemented solutions, tools, and strategies based on performance, feasibility, and industry standards.	L5
CO5	Design and develop a project/report demonstrating practical learning, problem-solving ability, and professional skills gained during the internship.	L6

Semester – VII

Course Code: CABO–1701

Credit: 4

Course Name: Soft Computing

Course Objectives:

- To introduce students to the principles and techniques of soft computing, which encompasses fuzzy logic, neural networks, evolutionary algorithms, and probabilistic reasoning.
- To teach methods for modeling and simulating complex systems that are tolerant to imprecision, uncertainty, and partial truth.
- To explore applications of soft computing in various domains such as pattern recognition, optimization, control systems, and decision-making.
- To provide hands-on experience with soft computing tools and algorithms through practical exercises and projects.
- To foster critical thinking and problem-solving skills by analyzing and designing soft computing solutions for real-world problems.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of soft computing, including fuzzy logic, neural networks, evolutionary computing, and their characteristics and applications.	L2
CO2	Apply soft computing techniques such as fuzzy inference systems, artificial neural networks, genetic algorithms, and swarm intelligence methods to solve real-world problems.	L3
CO3	Analyze complex problems using hybrid soft computing models, probabilistic reasoning, and rough set theory for decision-making and optimization.	L4
CO4	Evaluate the performance of different soft computing approaches and models based on accuracy, efficiency, interpretability, and ethical considerations.	L5
CO5	Design and develop intelligent systems using advanced soft computing techniques, including deep learning and explainable AI, for applications in domains like healthcare, finance, robotics, and bioinformatics.	L6

Block I: Introduction to Soft Computing

Unit 1: Introduction to Soft Computing: Definition and scope of Soft Computing, Characteristics of Soft Computing techniques, Applications and advantages of Soft Computing

Unit 2: Fuzzy Logic: Introduction to fuzzy sets and fuzzy logic, Fuzzy set operations, Fuzzy inference systems, Applications of fuzzy logic in control systems and decision making

Unit 3: Neural Networks: Introduction to artificial neural networks (ANNs), Single-layer and multi-layer perceptrons, Training algorithms: Backpropagation, Gradient Descent, Applications of neural networks in pattern recognition and classification

Unit 4: Evolutionary Computing: Introduction to evolutionary algorithms (Genetic Algorithms, Genetic Programming, Evolutionary Strategies), Principles of evolution and natural selection, Genetic operators: selection, crossover, mutation, Applications of evolutionary computing in optimization and machine learning.

Block II: Advanced Soft Computing Techniques

Unit 5: Hybrid Soft Computing Systems: Integration of fuzzy logic, neural networks, and evolutionary computing, Neuro-fuzzy systems, Genetic fuzzy systems, Applications and advantages of hybrid systems

Unit 6: Swarm Intelligence: Introduction to swarm intelligence, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Applications of swarm intelligence in optimization problems

Unit 7: Rough Sets: Introduction to rough sets theory, Rough set operations, Rough set-based feature selection, Applications of rough sets in data analysis and decision support systems

Unit 8: Probabilistic Reasoning: Introduction to Bayesian networks, Probabilistic inference, Learning Bayesian networks from data, Applications of probabilistic reasoning in uncertainty modeling

Block III: Applications of Soft Computing

Unit 9: Soft Computing in Image Processing: Fuzzy image processing techniques, Neural networks for image classification and recognition, Evolutionary algorithms for image enhancement, Applications in medical imaging, remote sensing, and computer vision

Unit 10: Soft Computing in Robotics: Fuzzy logic-based control systems for robots, Neural network-based robot learning and control, Evolutionary robotics, Applications in industrial automation, autonomous vehicles, and human-robot interaction

Unit 11: Soft Computing in Finance: Fuzzy logic-based financial modeling and forecasting, Neural networks for stock market prediction, Evolutionary algorithms for portfolio optimization, Applications in risk management, trading strategies, and investment decision making

Unit 12: Soft Computing in Bioinformatics: Fuzzy clustering for gene expression analysis, Neural networks for protein structure prediction, Evolutionary algorithms for sequence alignment, Applications in genomics, proteomics, and drug discovery

Block IV: Emerging Trends in Soft Computing

Unit 13: Deep Learning: Introduction to deep neural networks (DNNs), Convolutional Neural Networks (CNNs) for image recognition, Recurrent Neural Networks (RNNs) for sequential data, Applications and recent advances in deep learning

Unit 14: Explainable AI (XAI): Importance of interpretability in AI systems, Techniques for explaining decisions made by AI models, Interpretable models: decision trees, rule-based systems, Applications in healthcare, finance, and law

Unit 15: Quantum Computing and Soft Computing: Introduction to quantum computing, Quantum-inspired algorithms for optimization, Hybrid quantum-classical approaches in soft computing, Potential applications and future prospects

Unit 16: Ethics and Social Implications of Soft Computing: Ethical considerations in AI and soft computing, Bias and fairness in AI systems, Privacy concerns and data protection, Societal impact and responsible AI development

References:

1. "Soft Computing: Techniques and its Applications in Electrical Engineering" by D.K. Chaturvedi, S.P. Singh, and V.K. Sharma (2008)
2. "Soft Computing and Intelligent Systems: Theory and Applications" by Madan M. Gupta and Narsingh Deo (2000)
3. "Introduction to Soft Computing" by Eva Volná, JánŠtepán, and V ladik Kreinovich (2012)
4. "Soft Computing: Techniques and Applications" edited by K. Vijayakumar, V. Manickavasagam, and M. Marikkannu(2012)
5. "Handbook of Soft Computing for Engineers and Scientists" edited by F. CABOtillo, P. Melin, and O. Montiel (2013)

Course Code: CABO–1702

Credit: 4

Course Name: R programming

Course Objectives:

- To introduce students to the R programming language and its applications in statistical computing and data analysis.
- To teach fundamental concepts such as data types, functions, loops, and conditional statements in R.
- To explore advanced topics including data manipulation, visualization, and statistical modeling using R packages such as dplyr, ggplot2, and lm.
- To provide hands-on experience with real-world datasets through practical exercises and projects.
- To prepare students to perform data analysis and statistical inference tasks using R, and to communicate results effectively through visualizations and reports.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of R programming, including syntax, data types, data structures, control statements, and data handling techniques.	L2
CO2	Apply R programming concepts and tools such as RStudio and packages like ggplot2 and dplyr to perform data import, cleaning, manipulation, and visualization.	L3
CO3	Analyze datasets using descriptive statistics, data transformation techniques, and visualization methods to extract meaningful insights.	L4
CO4	Evaluate statistical models, visualization techniques, and R programming approaches based on accuracy, efficiency, and suitability for different data analysis tasks.	L5
CO5	Design and develop data analysis solutions and statistical models using advanced R programming techniques, including functional programming and package integration.	L6

Block I: Introduction to R Programming

Unit 1: Introduction to R: Overview of R programming language, Installation and setup of R and RStudio, Basic syntax, data types, and variables,

Unit 2: Data Structures in R: Vectors, matrices, arrays, and data frames, Creating and manipulating data structures, Indexing and subsetting data objects,

Unit 3: Introduction to R Studio: R studio, Installation, set up directory, Function of R studio, Advantages and disadvantages

Unit 4: Control Structures: Conditional statements (if-else), Loops (for, while), Functions and user-defined functions

Unit 5: Data Import and Export: Reading and writing data from/to external files (CSV, Excel, text), Using built-in functions and packages for data manipulation, Handling missing data and data cleaning techniques

Block II: Data Visualization with R

Unit 6: Introduction to Data Visualization: Importance of data visualization, Overview of plotting systems in R (base R, ggplot2)

Unit 7: Basic Plots in R: Creating scatter plots, line plots, and bar plots, Customizing plot aesthetics (colors, labels, legends), Adding titles, axes labels, and annotations to plots

Unit 8: Advanced Plots in R: Creating histograms, box plots, and density plots, Plotting multiple graphs on the same canvas, Using facetting and grouping for complex visualizations

Unit 9: Interactive Visualizations: Introduction to interactive plotting libraries (plotly, ggplotly), Creating interactive plots with tooltips, zooming, and brushing, Exporting interactive plots for web applications and presentations

Block III: Data Manipulation and Analysis

Unit 10: Data Manipulation with dplyr: Introduction to the dplyr package, Filtering, sorting, and summarizing data, Group-wise operations and aggregations

Unit 11: Data Transformation with tidyr: Reshaping data with tidyr functions (gather, spread), Handling missing values and data imputation, Combining and merging datasets

Unit 12: Descriptive Statistics: Calculating summary statistics (mean, median, variance), Computing correlations and covariance, Generating frequency tables and cross-tabulations

Block IV: Advanced R Programming Techniques

Unit 13: Statistical Modeling Basics: Introduction to statistical modeling concepts, Linear regression analysis, Hypothesis testing and p-values

Unit 14: Introduction to R Packages: Overview of R package structure and development, Installing and loading R packages, Using popular packages for specific tasks (e.g., data manipulation, visualization)

Unit 15: Error Handling and Debugging, Identifying and troubleshooting common errors in R code, Using debugging tools (debug, browser), Implementing error handling techniques (tryCatch, stop)

Unit 16: Functional Programming in R: Understanding functional programming concepts, Using apply functions (apply, lapply, sapply), Writing and applying custom functions

References:

1. "R for Data Science" by Hadley Wickham and Garrett Grolemund(2017)
2. "Advanced R" by Hadley Wickham (2019)
3. "The Art of R Programming: A Tour of Statistical Software Design" by Norman Matloff (2011)

4. "R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics" by Paul Teetor (2011)
5. "Introduction to Statistical Learning with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani(2021)
6. "ggplot2: Elegant Graphics for Data Analysis" by Hadley Wickham (2023)

Course Code: CABO–1703

Credit: 4

Course Name: Network Security & Cryptography

Course Objectives:

- To introduce students to the principles and concepts of network security, including threats, vulnerabilities, and countermeasures.
- To teach cryptographic techniques such as encryption, decryption, digital signatures, and hash functions.
- To explore protocols and algorithms used in secure communication, such as SSL/TLS, IPsec, and SSH.
- To provide hands-on experience with network security tools and techniques for intrusion detection, firewall configuration, and secure network design.
- To foster critical thinking and problem-solving skills by analyzing security threats and designing secure systems and protocols.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of network security, including security goals (CIA triad), threats, attacks, cryptographic principles, and security policies.	L2
CO2	Apply cryptographic algorithms and protocols such as symmetric and asymmetric encryption, hashing, and secure communication protocols like SSL/TLS and IPsec to ensure data security.	L3
CO3	Analyze various network attacks, vulnerabilities, and security mechanisms including firewalls, IDS/IPS, VPNs, and wireless security techniques.	L4
CO4	Evaluate security solutions, authentication methods, and risk management strategies based on effectiveness, scalability, and compliance with legal and ethical standards.	L5
CO5	Design and develop comprehensive network security frameworks, including incident response plans, disaster recovery strategies, and secure system architectures for real-world environments.	L6

Block I: Introduction to Network Security

Unit 1: Introduction to Network Security: Overview of network security concepts and principles, Importance of network security in modern computing environments, Goals of network security: confidentiality, integrity, availability (CIA triad)

Unit 2: Threats and Attacks: Common network threats and vulnerabilities, Types of network attacks: eavesdropping, interception, spoofing, denial of service (DoS), distributed denial of service (DDoS), etc., CABOe studies of notable network security breaches and incidents

Unit 3: Security Policies and Mechanisms: Role of security policies in enforcing security requirements, Access control mechanisms: authentication, authorization, and accounting (AAA), Cryptography as a fundamental mechanism for ensuring data confidentiality and integrity

Unit 4: Cryptography Basics: Introduction to cryptography and its role in network security, Basic concepts: encryption, decryption, plaintext, ciphertext, keys, Types of cryptographic algorithms: symmetric encryption, asymmetric encryption, hashing,

Block II: Cryptographic Protocols and Algorithms

Unit 5: Symmetric Encryption: Principles of symmetric encryption algorithms (e.g., DES, AES), Modes of operation: ECB, CBC, CFB, OFB, Key management and distribution in symmetric encryption,

Unit 6: Asymmetric Encryption: Principles of asymmetric encryption algorithms (e.g., RSA, ElGamal), Key exchange protocols: Diffie-Hellman key exchange, Digital signatures and public key infrastructure (PKI)

Unit 7: Hash Functions and Message Authentication Codes (MACs): Overview of hash functions and their properties, Applications of hash functions: data integrity, password hashing, Message Authentication Codes (MACs) for data authentication and integrity verification

Unit 8: Cryptographic Protocols: Secure communication protocols: SSL/TLS, IPsec, Cryptographic protocols for key exchange, authentication, and secure messaging, CABOe studies of protocol vulnerabilities and attacks

Block III: Network Security Mechanisms

Unit 9: Access Control and Authentication Mechanisms: Role-based access control (RBAC) and access control lists (ACLs), Authentication mechanisms: passwords, biometrics, multifactor authentication, Single sign-on (SSO) solutions and federated identity management

Unit 10: Firewalls and Intrusion Detection Systems (IDS): Overview of firewalls: packet filtering, stateful inspection, application-layer filtering, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Deployment strategies and best practices for firewalls and IDS/IPS

Unit 11: Virtual Private Networks (VPNs) and Secure Remote Access: Introduction to VPN technologies: IPsec VPNs, SSL VPNs, Secure remote access solutions: VPN clients, remote desktop protocols, Considerations for implementing secure remote access for organizations

Unit 12: Wireless Network Security: Common security threats and vulnerabilities in wireless networks, Security mechanisms for Wi-Fi networks: WPA2, WPA3, Best practices for securing wireless LANs and mobile devices

Block IV: Security Management and Incident Response

Unit 13: Security Management Practices: Risk management and threat assessment methodologies, Security policies, standards, and procedures, Security awareness training and employee education programs

Unit 14: Incident Response and Disaster Recovery: Incident response planning and preparedness, Incident detection, analysis, and containment procedures, Business continuity planning and disaster recovery strategies

Unit 15: Legal and Ethical Aspects of Network Security: Legal frameworks and regulations governing network security (e.g., GDPR, HIPAA), Ethical considerations in network security practices, CABOe studies of legal and ethical issues in network security

Unit 16: Emerging Trends and Future Directions: Current trends in network security: cloud security, IoT security, AI-driven security, Research challenges and opportunities in network security, Strategies for staying updated and adapting to evolving threats in network security

References:

1. "Computer Security: Principles and Practice" by William Stallings and Lawrie Brown (2018)
2. "Cryptography and Network Security: Principles and Practice" by William Stallings (2018)
3. "Network Security Essentials: Applications and Standards" by William Stallings (2021)
4. "Introduction to Cryptography with Coding Theory" by Wade Trappe and Lawrence C. Washington (2020)
5. "Principles of Computer Security: CompTIA Security+ and Beyond" by Wm. Arthur Conklin, Greg White, Chuck Cothren, Roger L. Davis, and Dwayne Williams (2022)

Course Code: CABO–1704

Credit: 4

Course Name: Parallel Computing

Course Objectives:

- To introduce students to the principles and techniques of parallel computing, which involve executing multiple computations simultaneously.
- To teach parallel programming models, including shared memory, distributed memory, and hybrid models.
- To explore parallel algorithms and data structures optimized for parallel execution, such as parallel sorting, matrix multiplication, and graph algorithms.
- To provide hands-on experience with parallel computing frameworks and libraries, such as OpenMP, MPI, CUDA, and Hadoop.
- To foster critical thinking and problem-solving skills by analyzing scalability, load balancing, and synchronization issues in parallel computing systems.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of parallel computing, including architectures, programming paradigms, platforms, and performance metrics such as speedup and scalability.	L2
CO2	Apply parallel programming models and tools such as MPI, OpenMP, and CUDA to implement parallel algorithms.	L3
CO3	Analyze parallel algorithms for matrix operations, sorting, searching, and numerical methods with respect to efficiency, load balancing, and scalability.	L4
CO4	Evaluate the performance of parallel systems and programs using profiling tools and optimization techniques, considering I/O, memory usage, and computational efficiency.	L5
CO5	Design and develop optimized parallel solutions and hybrid models for real-world problems, including machine learning applications and high-performance computing systems.	L6

Block I: Introduction to Parallel Computing

Unit 1: Overview of Parallel Computing: Definition and significance of parallel computing, Historical development and evolution of parallel computing, Motivation for parallelism: Performance enhancement, scalability, and solving larger problems

Unit 2: Parallel Computing Architectures: Classification of parallel computing architectures: Shared memory, distributed memory, hybrid architectures, SIMD vs. MIMD architectures, Characteristics, advantages, and limitations of each architecture type

Unit 3: Parallel Programming Paradigms: Introduction to parallel programming models: Message Passing Interface (MPI), OpenMP, CUDA, OpenCL, Overview of data parallelism, task parallelism, and hybrid parallelism, Comparison of different parallel programming paradigms

Unit 4: Parallel Computing Platforms and Tools: Overview of parallel computing platforms: Multi-core processors, GPUs, clusters, supercomputers, Introduction to parallel computing tools and libraries: MPI libraries, CUDA toolkit, OpenMP directives, Setting up development environments for parallel programming

Block II: Parallel Algorithms and Techniques

Unit 5: Parallel Algorithm Design: Principles of parallel algorithm design: Decomposition, mapping, and scheduling, Strategies for identifying parallelism in algorithms: Task parallelism, data parallelism, pipelining, Analysis of parallel algorithm efficiency: Speedup, efficiency, scalability

Unit 6: Parallel Matrix and Vector Operations: Parallel algorithms for basic linear algebra operations: Matrix multiplication, vector addition, Parallelization techniques for matrix and vector computations, Performance optimization strategies for parallel matrix and vector operations

Unit 7: Parallel Sorting and Searching Algorithms: Parallel sorting algorithms: Bitonic sort, parallel merge sort, parallel quicksort, Parallel searching algorithms: Binary search, parallel search on trees and graphs, Analysis of parallel sorting and searching algorithms

Unit 8: Parallelization of Numerical Methods: Parallelization of numerical methods: Finite difference methods, finite element methods, Monte Carlo methods, Parallel algorithms for solving linear and nonlinear equations, numerical integration, and differential equations, Challenges and considerations in parallelizing numerical methods

Block III: Parallel Programming Models and Languages

Unit 9: Introduction to MPI Programming: Basics of message passing programming model, MPI concepts: Communicators, ranks, point-to-point communication, collective communication, Writing and executing MPI programs

Unit 10: Introduction to OpenMP Programming: Basics of shared memory programming model, OpenMP directives and constructs for parallelism: Parallel regions, work-sharing constructs, synchronization, Writing and executing OpenMP programs

Unit 11: GPU Programming with CUDA: Introduction to GPU architecture and CUDA programming model, CUDA programming basics: Kernel execution, memory management, thread synchronization, Writing and executing CUDA programs

Unit 12: Hybrid Programming with MPI and OpenMP: Combining message passing and shared memory parallelism, Strategies for hybrid parallel programming: Task distribution, load balancing, data partitioning, Writing and executing hybrid MPI+OpenMP programs

Block IV: Performance Optimization and Advanced Topics

Unit 13: Performance Analysis and Profiling: Techniques for performance measurement and analysis: Execution time, speedup, scalability, Profiling tools for identifying performance bottlenecks: gprof, Valgrind, NVIDIA Visual Profiler, Optimization strategies based on performance analysis

Unit 14: Parallel I/O and Storage: Parallel I/O concepts and challenges, Techniques for parallelizing I/O operations: Parallel file systems, I/O buffering, data striping, Strategies for efficient storage and retrieval of data in parallel computing environments

Unit 15: Parallelization of Machine Learning Algorithms: Parallel machine learning algorithms: Parallelization of training and inference processes, Parallel computing frameworks for machine learning: TensorFlow, PyTorch, Apache Spark MLlib, Case studies of parallel machine learning applications

Unit 16: Emerging Trends in Parallel Computing: Overview of emerging technologies and trends in parallel computing: Quantum computing, neuromorphic computing, edge computing, Challenges and opportunities in future parallel computing research and development, Discussion on the impact of parallel computing on various domains: Scientific computing, data analytics, artificial intelligence

References:

1. "Introduction to Parallel Computing" by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar (2021)
2. "Parallel Programming: Concepts and Practice" by Barry Wilkinson and Michael Allen (2018)
3. "Parallel Computing: Principles and Practice" by Rob H. Bisseling(2004)
4. "Parallel Programming in C with MPI and OpenMP" by Michael J. Quinn(2004)
5. "CUDA by Example: An Introduction to General-Purpose GPU Programming" by Jason Sanders and Edward Kandrot (2010)

Course Code: CABO-1705

Credit: 4

Course Name: Linear Programming

Course Objectives:

- To introduce students to the theory and applications of linear programming, which involves optimizing a linear objective function subject to linear constraints.
- To teach the formulation of mathematical models as linear programming problems, including decision variables, objective functions, and constraints.
- To explore methods for solving linear programming problems, such as the simplex method, interior point methods, and graphical methods.
- To provide hands-on experience with linear programming software packages such as Excel Solver, GNU Linear Programming Kit (GLPK), and MATLAB Optimization Toolbox.
- To foster critical thinking skills by applying linear programming techniques to real-world optimization problems in various domains such as operations research, economics, engineering, and logistics.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of linear programming problems, including formulation, convex sets, extreme points, and graphical solution methods.	L2
CO2	Apply graphical and simplex-based methods (Simplex, Big-M, and Two-Phase) to solve linear programming problems.	L3
CO3	Analyze linear programming models using duality theory, dual-simplex method, and assignment problem techniques such as the Hungarian method.	L4
CO4	Evaluate optimal solutions of transportation and assignment problems using methods like MODI, while handling special cases such as degeneracy.	L5
CO5	Design and formulate real-world optimization problems (transportation, assignment, and resource allocation) and develop appropriate solution strategies.	L6

Block I: Linear Programming Problems

Unit 1: Linear programming problem and its formulation

Unit 2: Geometric interpretation, Convex set, Convex Hull and related results

Unit 3: Extreme points and Basic feasible solutions

Unit 4: Graphical Method to solve LPP

Block II: The Simplex Algorithm

Unit 5: Simplex Method

Unit 6: Big-M Method

Unit 7: Two-Phase Method

Unit 8: Special CABOes in Simplex Method

Block III: Duality Theory and Assignment Problem

Unit 9: Duality in LPP

Unit 10: Dual-Simplex Method

Unit 11: Mathematical formulation of assignment problem

Unit 12: Hungarian Method

Block IV: Transportation Problem

Unit 13: Mathematical formulation of transportation problem

Unit 14: Solution of transportation problem

Unit 15: MODI Method

Unit 16: Degeneracy in transportation problems

References:

1. "Operations Research", S. D. Sharma, Kedarnath Ramnath & Company.(2010)
2. "Linear Programming", G. Hadley, Oxford, IBH publishing Co.(2024)
3. "Operations Research", P.K. Gupta, S. Chand & Company Ltd.(2015)
4. Elements of Operations Research" by Dr. P.K. Gupta, published by Sultan Chand & Sons in (2022).
5. "Problems in Operations Research" by Dr. P.K. Gupta and Man Mohan, also published by Sultan Chand & Sons in (2014).

Semester – VIII

Course Code: CABO–1801

Credit: 4

Course Name: Natural Language Processing

Course Objectives:

- To introduce students to the field of natural language processing, which involves the interaction between computers and human languages.
- To teach fundamental concepts such as tokenization, stemming, lemmatization, part-of-speech tagging, and syntactic parsing.
- To explore techniques for text classification, sentiment analysis, named entity recognition, and machine translation.
- To provide hands-on experience with NLP libraries and tools such as NLTK (Natural Language Toolkit), spaCy, and TensorFlow.
- To foster critical thinking and problem-solving skills by applying NLP techniques to real-world applications such as chatbots, information retrieval, and text summarization.

Course Outcomes

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of Natural Language Processing, including text processing, linguistic features, tokenization, and regular expressions.	L2
CO2	Apply NLP techniques such as text preprocessing, feature extraction (BoW, TF-IDF), and basic machine learning algorithms to perform tasks like text classification and named entity recognition.	L3
CO3	Analyze textual data using advanced NLP methods such as word embeddings, sentiment analysis, topic modeling, and sequence models to extract meaningful insights.	L4
CO4	Evaluate the performance of NLP models and techniques, including classification, machine translation, and language models, using appropriate evaluation metrics and ethical considerations.	L5
CO5	Design and develop NLP-based applications such as chatbots, speech recognition systems, and intelligent text processing solutions using modern frameworks and transformer-based models.	L6

Block I: Introduction to Natural Language Processing

Unit 1: Overview of NLP: Definition and history of NLP, Applications of NLP, Basic concepts and terminology

Unit 2: Text Processing and Tokenization: Text normalization (lowerCABOing, stemming, lemmatization), Tokenization techniques, Handling punctuation, numbers, and special characters,

Unit 3: Linguistic Fundamentals for NLP: Syntax and semantics, Parts of speech (POS) tagging, Parsing (dependency and constituency parsing)

Unit 4: Regular Expressions and Text Matching: Introduction to regular expressions, Common text matching patterns, Use CABOes in text processing

Block II: Core NLP Techniques and Algorithms

Unit 5: Bag-of-Words and TF-IDF, Concept of Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Feature extraction from text

Unit 6: Word Embeddings and Vector Space Models: Introduction to word embeddings, Word2Vec, GloVe, FastText, Contextual embeddings (ELMo, BERT)

Unit 7: Named Entity Recognition (NER): Concept of named entities, NER algorithms and models, Evaluation metrics for NER

Unit 8: Text Classification: Supervised learning for text classification, Common algorithms (Naive Bayes, SVM, Logistic Regression), Evaluation metrics (accuracy, precision, recall, F1-score)

Block III: Advanced NLP Techniques

Unit 9: Sentiment Analysis: Introduction to sentiment analysis, Rule-based vs. machine learning approaches, Applications in social media and customer feedback

Unit 10: Topic Modeling: Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), Applications and visualization of topics

Unit 11: Sequence Models and Language Models: Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), Transformer models and attention mechanisms

Unit 12: Machine Translation: History and evolution of machine translation, Statistical vs. neural machine translation, Evaluation of translation quality (BLEU score)

Block IV: Practical Applications and Emerging Trends

Unit 13: Speech Recognition and Processing: Basics of speech recognition, Acoustic models, language models, and pronunciation models, Current technologies and applications

Unit 14: Dialogue Systems and Chatbots: Structure of dialogue systems, Intent recognition and slot filling, Designing and building chatbots

Unit 15: Ethical Considerations in NLP: Bias and fairness in NLP models, Privacy concerns and data protection, Ethical AI and responsible use of NLP technologies

Unit 16: Recent Trends and Future Directions: Advances in pre-trained language models (e.g., GPT-3, T5), Zero-shot and few-shot learning, Future research directions and open challenges in NLP, Supplementary Materials

References:

1. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" by Daniel Jurafsky and James H. Martin (2025)
2. "Natural Language Processing in Action" by Lane, H., Howard, H., and Hapke, H. (2019)
3. "Speech and Language Processing: Algorithms and Systems" by Mohamed Zakaria Kurdi and Khalid Shaalan(2021)
4. "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain, and Karan Kumar (2018)
5. "Applied Natural Language Processing: Identification, Investigation, and Resolution" by John P. McCrae and Sergio Roa(2022)

Course Code: CABO–1802

Credit: 4

Course Name: Web Designing

Course Objectives

- To introduce students to the principles and techniques of web design, including layout, typography, color theory, and user experience (UX) design.
- To teach HTML (Hypertext Markup Language) and CSS (CABOCading Style Sheets) for creating and styling web pages.
- To explore responsive web design principles for creating websites that adapt to different screen sizes and devices.
- To provide hands-on experience with web design tools and frameworks such as Adobe Dreamweaver, Bootstrap, and WordPress.
- To foster creativity and critical thinking skills by designing and implementing web projects that meet user requirements and usability standards.

Course Outcomes

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of web design, including HTML, CSS, JavaScript, web architecture, responsive design, and accessibility principles.	L2
CO2	Apply HTML5, CSS3, and JavaScript (including ES6 features) along with frameworks like Bootstrap to develop responsive and interactive web pages.	L3
CO3	Analyze web design requirements and user needs to select appropriate layouts, frameworks, performance optimization techniques, and accessibility standards.	L4
CO4	Evaluate different web technologies, frameworks (such as React, Angular), and design strategies based on usability, performance, SEO, and scalability.	L5
CO5	Design and develop complete, user-friendly, and optimized web applications incorporating modern tools, responsive design, accessibility, and emerging technologies like PWAs and SPAs.	L6

Block I: Foundations of Web Design

Unit 1: Introduction to Web Design: History and evolution of the web, Understanding the role of a web designer, Overview of web technologies (HTML, CSS, JavaScript)

Unit 2: HTML Basics: Structure of an HTML document, Common HTML tags (headings, paragraphs, links, images, lists), Semantic HTML and accessibility considerations, Forms and input elements

Unit 3: CSS Fundamentals: CSS syntax and selectors, Styling text and fonts, Box model: margins, padding, borders, Layout techniques: float, flexbox, grid

Unit 4: Introduction to JavaScript: Basics of JavaScript: syntax, variables, data types, DOM manipulation, Event handling, Introduction to ES6 features (let, const, arrow functions)

Block II: Advanced HTML, CSS, and JavaScript

Unit 5: Advanced HTML Techniques: HTML5 elements (video, audio, canvas), Data attributes and microdata, Responsive design principles, Introduction to frameworks (Bootstrap)

Unit 6: Advanced CSS Techniques: Advanced selectors and combinators, CSS animations and transitions, Preprocessors (Sass, LESS), CSS methodologies (BEM, OOCSS)

Unit 7: Advanced JavaScript and ES6+: Advanced JavaScript concepts (closures, promises, async/await), JavaScript frameworks and libraries overview (React, Vue, Angular), AJAX and Fetch API, Local storage and session storage

Unit 8: Responsive Web Design: Principles of responsive design, Media queries, Mobile-first design, Testing and debugging responsive layouts

Block III: Tools and Frameworks for Web Design

Unit 9: Web Design Tools and Workflow: Design tools (Adobe XD, Sketch, Figma), Prototyping and wireframing, Version control with Git, Command-line basics and task runners (Gulp, Grunt)

Unit 10: Front-End Frameworks and Libraries: Introduction to Bootstrap, Material Design principles and implementation, Foundation framework, Utilizing icon libraries (Font Awesome, Material Icons)

Unit 11: CSS Frameworks and Preprocessors: Benefits of using CSS frameworks, Sass: syntax, variables, nesting, mixins, functions, LESS: syntax and features, Best practices for using preprocessors

Unit 12: JavaScript Frameworks and Libraries: Introduction to React: components, state, props, Vue.js: reactivity system, directives, components, Angular: modules, components, services, Comparing and choosing the right framework

Block III: Practical Applications and Emerging Trends

Unit 13: Web Performance Optimization: Importance of web performance, Techniques for optimizing images and media, Minimizing and bundling resources (minification, concatenation), Tools for performance testing (Lighthouse, WebPageTest)

Unit 14: Web Accessibility: Principles of web accessibility (WCAG guidelines), ARIA (Accessible Rich Internet Applications), Tools for testing accessibility, Implementing accessible forms and navigation

Unit 15: SEO and Web Analytics: Basics of SEO (Search Engine Optimization), On-page and off-page SEO techniques, Introduction to web analytics, Using Google Analytics for tracking and reporting

Unit 16: Emerging Trends and Future Directions: Progressive Web Apps (PWAs), Single Page Applications (SPAs), WebAssembly, Future of web design and development

References:

1. Robbins, J. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics* (5th ed.). O'Reilly Media. ISBN: 978-1491960202.
2. Meyer, E. A., & Weyl, E. (2017). *CSS: The Definitive Guide* (4th ed.). O'Reilly Media. ISBN: 978-1449393199.
3. Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley. ISBN: 978-1118531648.
4. Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Wiley. ISBN: 978-1118008188.
5. Crockford, D. (2008). *JavaScript: The Good Parts*. O'Reilly Media. ISBN: 978-0596517748.

Course Code: CABO–1803

Credit: 4

Course Name: Data Warehousing and Data Mining

Course Objectives

- To introduce students to the concepts and technologies of data warehousing, which involves the collection, storage, and management of large volumes of data from various sources.
- To teach data mining techniques for extracting valuable insights and patterns from data stored in data warehouses.
- To explore data preprocessing methods such as data cleaning, transformation, and integration to prepare data for analysis.
- To provide hands-on experience with data mining algorithms and tools such as association rule mining, classification, clustering, and anomaly detection.
- To foster critical thinking skills by applying data warehousing and data mining techniques to real-world datasets and business problems.

Course Outcomes

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of data warehousing and data mining, including architecture, ETL processes, OLAP systems, and knowledge discovery techniques.	L2
CO2	Apply data preprocessing methods, data warehousing design techniques, and data mining algorithms using tools like WEKA, RapidMiner, and KNIME.	L3
CO3	Analyze datasets using classification, clustering, association rule mining, and advanced data mining techniques to extract meaningful patterns and insights.	L4
CO4	Evaluate different data warehousing models, mining algorithms, and analytical approaches based on performance, scalability, accuracy, and ethical considerations.	L5
CO5	Design and develop integrated data warehousing and data mining solutions for real-world applications such as fraud detection, recommendation systems, and business intelligence.	L6

Block I: Introduction to Data Warehousing

Unit 1: Fundamentals of Data Warehousing: Definition and Importance, Evolution and History, Data Warehousing Concepts, Difference between Database and Data Warehouse

Unit 2: Data Warehousing Architecture: Basic Architecture of Data Warehousing, Data Warehouse Models (Enterprise Warehouse, Data Mart, Virtual Warehouse), ETL (Extract, Transform, Load) Processes, Data Integration and Data Loading Techniques

Unit 3: Data Warehouse Design and Implementation: Designing a Data Warehouse: Methodologies and Approaches, Data Warehouse Schema: Star, Snowflake, and Galaxy, Data Partitioning Strategies, Indexing and Query Optimization

Unit 4: Data Warehouse Tools and Technologies: Overview of Data Warehousing Tools, OLAP (Online Analytical Processing) Systems, MOLAP, ROLAP, and HOLAP, CABOe Studies of Data Warehousing Tools (e.g., Microsoft SQL Server, Oracle)

Block II: Introduction to Data Mining

Unit 5: Fundamentals of Data Mining: Definition and Importance, Knowledge Discovery in Databases (KDD) Process, Data Mining vs. Query Tools, Data Mining Tasks: Descriptive and Predictive

Unit 6: Data Preprocessing: Data Cleaning, Data Integration, Data Reduction, Data Transformation and Discretization

Unit 7: Data Mining Techniques I: Classification: Decision Trees, Naive Bayes, k-Nearest Neighbors, Regression Analysis, Evaluation of Classifiers: Accuracy, Precision, Recall, F-measure

Unit 8: Data Mining Techniques II: Clustering: k-Means, Hierarchical Clustering, DBSCAN, Association Rule Mining: Apriori Algorithm, FP-Growth, Sequence and Time Series Mining

Block III: Advanced Data Mining and Applications

Unit 9: Advanced Topics in Data Mining: Text Mining and Web Mining, Spatial and Temporal Data Mining, Graph and Network Mining, Big Data Mining

Unit 10: Data Mining Tools and Technologies: Overview of Data Mining Tools (e.g., WEKA, RapidMiner, KNIME), Open Source Data Mining Software, Implementing Data Mining Algorithms in R and Python

Unit 11: Data Mining Applications I: Market Basket Analysis, Fraud Detection, Customer Relationship Management (CRM), Healthcare and Bioinformatics

Unit 12: Data Mining Applications II: Social Media and Sentiment Analysis, Recommender Systems, Financial Data Analysis, Telecommunications and Network Security

Block IV: Integration and Future Trends

Unit 13: Deep Learning for Data Mining: Privacy-preserving techniques in deep learning: federated learning, secure model aggregation, Secure deployment of deep learning models in production environments, Adversarial robustness in deep learning models for protecting against attacks

Unit 14: Integration of Data Warehousing and Data Mining: Synergies between Data Warehousing and Data Mining, Real-time Data Warehousing and Mining, Data Warehouse as a Data Source for Data Mining, CABOe Studies and Examples

Unit 15: Ethical and Privacy Issues: Data Privacy and Security Concerns, Ethical Implications of Data Mining, Data Anonymization Techniques, Legal Aspects and Regulations (e.g., GDPR)

Unit 16: Emerging Trends in Data Warehousing and Data Mining: Cloud Data Warehousing and Mining, Real-time Analytics, AI and Machine Learning Integration, Future Directions and Innovations

References:

1. "Security in Computing" by Charles P. Pfleeger and Shari Lawrence Pfleeger(2015)
2. "Building a Data Warehouse: With Examples in SQL Server" by Vincent Rainardi (2014)
3. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei (2011)
4. "Data Warehousing For Dummies" by Thomas C. Hammergren and Alan R. Simon (2009)
5. "Privacy-Preserving Data Mining: Models and Algorithms" by Charu C. Aggarwal and Philip S. Yu(2008)

Course Code: CABO–1804

Credit: 4

Course Name: Digital Forensics

Course Objectives:

- To introduce students to the principles and techniques of digital forensics, which involves the investigation and analysis of digital evidence for legal purposes.
- To teach methods for acquiring, preserving, and analyzing digital evidence from various sources such as computers, mobile devices, and networks.
- To explore forensic techniques for recovering deleted files, examining file metadata, and identifying traces of malicious activities.
- To provide hands-on experience with forensic tools and software used in digital investigations, such as EnCase, FTK (Forensic Toolkit), and Autopsy.
- To foster critical thinking and problem-solving skills by applying digital forensic techniques to real-world scenarios and EnCase studies involving cybercrime, data breaches, and intellectual property theft.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of digital forensics, investigation processes, types of digital evidence, file systems, and legal and ethical considerations.	L2
CO2	Apply digital forensic tools and techniques such as FTK Imager, Autopsy, and Volatility for evidence acquisition, data recovery, and forensic analysis.	L3
CO3	Analyze digital evidence from storage media, operating systems, and network traffic to investigate cyber incidents, malware attacks, and unauthorized activities.	L4
CO4	Evaluate different forensic methods, tools, intrusion detection systems, and incident response strategies based on effectiveness, reliability, and legal admissibility.	L5
CO5	Design and develop comprehensive digital forensic investigation solutions, including incident response plans, mobile and memory forensics strategies, and legal documentation for real-world cases.	L6

Block I: Introduction to Digital Forensics

Unit 1: Introduction to Digital Forensics: Definition and scope of digital forensics, Importance of digital evidence in investigations, Roles of Forensics Investigator, Forensics Readiness, Steps for Forensics Legal and ethical considerations in digital forensics

Unit 2: Computer Forensics Investigation Process: Digital Forensics Investigation Process, Digital Forensics Investigation Process-Assessment Phase, Acquire the Data, Analyze the Data, Report the Investigation

Unit 3: Digital Evidence Acquisition: Principles of digital evidence collection and preservation, Forensic imaging techniques for storage devices, Chain of custody and documentation procedures

Unit 4: Understanding Storage Media and File System: The Booting Process, LINUX Boot Process, Mac OS Boot Sequence, Windows 10 Booting Sequence, File System, Type of File Systems

Unit 5: File Systems and Data Recovery: Overview of file systems (FAT, NTFS, ext), Data recovery methods and tools, Carving techniques for extracting deleted files and fragments

Block II: Windows, Network Forensics and Incident Response

Unit 6: Windows Forensics: Introduction to Windows Forensics, Windows Forensics Volatile Information, Windows Forensics Non- Volatile Information, Recovering deleted files and partitions, Windows Forensics Summary,

Unit 7: Digital Forensics Road map: Static Data Acquisition from windows using FTK Imager, Live Data Acquisition using FTK Imager, FTK Imager, Installation of KALI Linux, RAM Dump Analysis using Volatility, Static Data Acquisition from Linux OS

Unit 8: Recovering Deleted Files and Partitions, Digital Forensics Tools, Overview of EnCABOe Forensics, Deep Information Gathering Tool: Dmitry Page, Use of Autopsy and FTK Imager

Unit 9: Introduction to Network Forensics: Basics of network protocols (TCP/IP, DNS, HTTP), Capturing and analyzing network traffic, Identifying and investigating network-based attacks

Block III: Forensic Investigation Techniques

Unit 10: Intrusion Detection and Prevention: Overview of intrusion detection systems (IDS) and intrusion prevention systems (IPS), Signature-based vs. anomaly-based detection techniques, Analyzing IDS/IPS logs for detecting and responding to security incidents

Unit 11: Malware Analysis: Types of malware (viruses, worms, Trojans, ransomware), Techniques for analyzing and reverse-engineering malware, Developing and applying signatures for malware detection

Unit 12: Incident Response and Handling: Incident response process and procedures, Establishing an incident response team (IRT), Incident containment, eradication, and recovery strategies

Block IV: Advanced Topics in Digital Forensics

Unit 13: Mobile Forensics: Introduction to mobile device forensics (smartphones, tablets), Extracting and analyzing data from mobile devices, Challenges and considerations in mobile forensics

Unit 14: Memory Forensics: Understanding volatile memory (RAM) and its forensic implications, Memory acquisition techniques and tools, Analyzing memory dumps for evidence extraction

Unit 15: Anti-Forensics Techniques; Common anti-forensics tactics used by attackers, Detecting and countering anti-forensics measures, Best practices for preserving digital evidence integrity

Unit 16: Legal Aspects of Digital Forensics: Digital evidence admissibility in court, Expert witness testimony and courtroom procedures, CABOe studies and landmark legal decisions in digital forensics

References:

1. "Guide to Computer Forensics and Investigations" by Bill Nelson, Amelia Phillips, and Christopher Steuart (2020)
2. "Computer Forensics and Cyber Crime: An Introduction" by Marjie T. Britz (2013)
3. "Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet" by Eoghan CABOey (2011)
4. "Handbook of Digital Forensics and Investigation" edited by Eoghan CABOey (2009)
5. "File System Forensic Analysis" by Brian Carrier (2005)

Course Code: CABO–1805

Credit: 4

Course Name: Bioinformatics Using AI

Course Objectives:

- To introduce students to the interdisciplinary field of bioinformatics, which involves the application of computational techniques to analyze biological data.
- To teach fundamental concepts in biology, genetics, and genomics relevant to bioinformatics analysis.
- To explore artificial intelligence (AI) techniques such as machine learning, deep learning, and natural language processing for analyzing biological data.
- To provide hands-on experience with AI tools and frameworks used in bioinformatics, such as TensorFlow, Keras, and scikit-learn.
- To foster critical thinking skills by applying AI techniques to analyze genomic sequences, predict protein structures, and discover biomarkers for disease diagnosis and treatment.

Course Outcomes:

On successful completion of this course, students will be able to:

CO1	Explain the fundamental concepts of bioinformatics, artificial intelligence, machine learning, deep learning, and their roles in computational research and biological data analysis.	L2
CO2	Apply data preprocessing techniques, feature extraction, dimensionality reduction, and machine learning algorithms (using tools like Weka) to analyze bioinformatics datasets.	L3
CO3	Analyze bioinformatics problems by selecting appropriate supervised, unsupervised, and ensemble learning methods for classification, regression, and clustering tasks.	L4
CO4	Evaluate the effectiveness of various machine learning and deep learning models, feature selection methods, and data handling techniques for accurate and efficient bioinformatics solutions.	L5
CO5	Design and develop AI-based models or systems for real-world bioinformatics applications such as structural prediction, data integration, and biological data interpretation.	L6

Block I: Introduction to Bioinformatics and AI

Unit 1: Overview of Bioinformatics and its applications in computational research, Bioinformatics data.

Unit2: Introduction to Artificial Intelligence (AI) fundamentals and applications, Fundamentals of AI and machine learning, Weka and machine learning algorithms.

Unit 3: Basics of computers for bioinformatics.

Unit 4: Fundamentals of machine learning and deep learning for data analysis.

Block II: Data Preprocessing and Feature Engineering

Unit 5: Data preprocessing techniques for computational data.

Unit 6: Feature extraction and selection methods in bioinformatics.

Unit 7: Dimensionality reduction techniques for high-dimensional data.

Unit 8: Handling missing data and noise in mathematical datasets.

Block III: Machine Learning Models in Bioinformatics

Unit 9: Supervised learning algorithms for bioinformatics (classification & regression).

Unit 10: Unsupervised learning techniques for clustering and pattern discovery in datasets.

Unit 11: Ensemble learning methods and their applications in bioinformatics.

Unit 12: Deep learning architectures for data modalities.

Block IV: Applications of AI in Bioinformatics

Unit 13: Structural bioinformatics and prediction methods with AI

Unit 14: Analysis of research data using AI techniques.

Unit 15: Data analysis and interpretation using machine learning and deep learning.

Unit 16: Integration of data and system modeling with AI approaches.

References:

1. Russell, S., & Norvig, P. (2021). Artificial intelligence: A modern approach. Pearson.
2. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Introduction to statistical learning: With applications in R. Springer.
4. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
5. Mount, D. W. (2004). Bioinformatics: Sequence and genome analysis. Cold Spring Harbor Laboratory Press.

MOOCs

The University shall give flexibility in opting for MOOC (Massive Online Open Courses)/ SWAYAM by the students pertaining to the prescribed curriculum and also the credits earned in the MOOC courses may be dealt as part of the evaluation scheme as per UGC (Online Learning Programme) Regulations, 2020.

Syllabi and Course Materials

The Syllabi, PPR and Self-Learning Materials (SLMs) are developed mostly by experienced faculty members of Usha Martin University in consultation with content experts and the same will be forwarded to CIQA and BOS/Academic Council/ Executive Council for further suggestions and approval.

Faculty Members

The University has identified the dedicated requisite faculty and support staff as mandated by the UGC and they are allocated the positions exclusively for OL mode. The course material prepared by the CDOE faculty is at par with the regulations 2020.

List of Faculty associated with BCA programme is as follows:

S. No.	Name of Faculty	Designation	Nature of Appointment	Qualification	Subject
1.	Dr. Sharmistha Roy	Associate Professor	Full-Time	Ph.D.	Computer Science

Delivery Mechanism

The OL of UMU follows a modern ICT (Information & Communication Technology) enabled approach for instruction. The methodology of instruction in OL of UMU is different from that of the conventional/regular programs. Our OL system is more learner-oriented and the learner is an active participant in the teaching-learning process. OL of UMU academic delivery system comprises:

A. Print Material

The printed material of the programme supplied to the students will be unit wise for every course.

B. Counselling Sessions

Normally, counselling sessions are held as per a schedule drawn beforehand by the Subject Coordinator. There will be 6 counselling/ contact classes for 4 credit courses will be held on the campus on Saturday and on Sunday of 2 hours duration for each course in face to face mode (In case of 2 credit course contact hours are required 6 hours and in case of 6 credit course contact hours required 18 hours). Contact classes will be held in the campus on Saturdays and on Sundays.

Attendance is mandatory for lab counseling sessions to ensure students gain the necessary practical skills. Students are required to attend at least 75% of lab sessions to be eligible for practical exams and assessments.

C. Medium of Instruction

Medium of Course Instruction: English

Medium of Examination: English

D. Student Support Systems

Universities Study Centers or Learner Support Centre shall be headed by a coordinator, not below the rank of Assistant professor and shall be augmented with academic and non-academic staff depending on the learner.

The university has made appropriate arrangements for various support services including counselling schedule and resource-oriented services evaluation methods and dates both online and offline modes for easy and smooth services to the students in online mode.

At present the university have only one study center on the campus. The institution is not promoting any study centers outside the campus. All student support services will be provided to the student through a single window method/mode onsite and online.

E. Procedure for Admissions, Curriculum, Transaction and Evaluation Admission Process

Admission to the BCA Programme will be done on the basis of screening of candidate's eligibility on first come first serve basis. The University will follow the reservation policy as per norms of the Government. Admission shall not be a right to the students and UMU, DDOE shall retain the right to cancel any admission at any point of time if any irregularity is found in the admission process, eligibility etc.

Maximum Duration

- A. The maximum duration of the BCA Programme is Eight years. Thereafter, students seeking completion of the left-over course(s) will be required to seek fresh admission.
- B. The student can complete his programme within a period of 8 years failing which he/she shall seek fresh admission to complete the programme.

Eligibility

10+2 from any recognized board.

Fee Structure

Name of the Program	Degree	Duration	Year	Tuition Fee/Year	Exam Fee/Year	Total (in Rs.)
Bachelor of Computer Science	UG	4 to 8 Years	1	36000	2000	38000
			2	36000	2000	38000
			3	36000	2000	38000
			4	36000	2000	38000
			Total			

Activity Schedule

S. No.	Name of the Activity	Tentative months schedule (specify months) during year			
		From(Month)	To (Month)	From(Month)	To (Month)
1	Admission	Jul	Sep	Jan	Mar
2	Assignment submission (if any)	Sep	Oct	Mar	Apr
3	Evaluation of assignment	Oct	Nov	Apr	May
4	Examination	Dec	Dec	Jun	Jun
5	Declaration of result	Jan	Jan	Jul	Jul
6	Re-registration	Jul	Jul	Jan	Jan
7	Distribution of SLM	Jul	Sep	Jan	Mar
8	Contact programmes (counselling, practical, etc.)	Sep	Nov	Mar	May

Credit System

UMU, CDOE proposes to follow the 'Credit System' for most of its programs. Each credit amounts to 30 hours of study comprising all learning activities. Thus, a 8 credit course requires 240 hours, 6 credit course requires 180 hours , 4 credit course requires 120 hours and 2 credit course requires 60 hours of study. This helps the student to understand the academic effort to complete a course. Completion of an academic programme requires successful clearing of both, the assignments and the term-end examination of each course in a programme.

Duration of the Programme	Credits	Name of the Programme	Level of the Programme
4 to 8 Years	160	BCA	Bachelor's Degree

Assignments

In order to ascertain the writing skill and level of comprehension of the learner, assignment work is compulsory for all learners. Each assignment shall consist of a number of questions, case studies and practical related tasks. The assignment question papers will be uploaded to the website within a scheduled time and the learners shall be required to respond them within a specified period of time. The response of the learner is examined by a faculty member.

Evaluation

The evaluation system of the programme is based on two components:

- A. Continuous evaluation in the form of assignments (weightage 30%):** This Component carries a weightage of 30%. There will be at least one graded assignment and test per course. These assignments are to be submitted to the Co-Ordinator of the programme to which the student is assigned or attached with.
- B. Term-end examination (weightage 70%):** This will be held twice every year in the months of June and December. The students are at liberty to appear in any of the examinations conducted by the University during the year. A student will be allowed to appear in the Term-End Examination only after she/he has registered for that course and submitted the assignment. For appearing in the Examination, every student has to submit an Examination form through online (<https://www.umu.ac.in>) / or offline before the due dates as given in the schedule of operations. If a student misses any term-end examination of a course for any reason, s/he may appear for any of them or all the courses subject to the maximum of 8 courses in the subsequent term-end examinations. This facility will be available until a student secures the minimum pass grade in the courses but up to a maximum period of four semesters, since the date of registration of the course is valid for four semesters. Beyond this period s/he may continue for another four semesters by getting Re-registration by paying fee again. In that case, the score of qualified assignments and/or term-end examination will be retained and the student will be required to complete the left out requirements of such re-registered courses. Minimum requirement for passing a course will be 40% marks.

Laboratory Support and Library Resources

The library of Usha Martin University aims to empower the teaching mission and intellectual culture of the community through availability through an organized collection of information as well as instruction in its access, relevance and evaluation.

The University Library enriches advance learning and discovery by providing access to a broad array of resources for education, research and creative work to ensure the rich interchange of ideas in the pursuit of knowledge.

The Usha Martin University has initiated the process of setting up a dedicated Library for OL program and acquiring printed books and e-books for this purpose. The required International and National subject journals are also provided. We already have annual journal subscriptions and the capacity can be enlarged at later stages as the University lines up with more online journals.

The collection of the Library is rich and diverse especially in terms of the breadth and depth of coverage. Collection encompasses subjects in Management, Commerce, Information Technology, Computer Applications, and other allied areas. This collection further includes Books, Research Journals, Project Reports/Dissertations and online Journals.

The University has well equipped Computer Laboratories, Lecture Capturing Systems, Audio Video facilities, ICT enabled class rooms, Wi-Fi facilities and Learning Management Systems etc.

Cost Estimate of the programme and the provisions

Initial expenses have been done by the University in terms of provision of infrastructure, manpower, printing of Self Study Material etc. The University intends to allocate expenses out of the total fee collection as per following details:

a) SLM Development and Distribution	:	20%
b) Postal and ICT Expenses	:	10%
c) Salary and other Administrative expenses	:	50%
d) Future Research development reserve	:	10%
e) Lab Instruments	:	10%

Once programmes are operational, the programme budget from fee receipts will be planned as per the guidelines of University Grants Commission.

Quality Assurance

The University has established the Centre for Internal Quality Assurance (CIQA) in the University campus. The CIQA will monitor and maintain the quality of the OL programmes. It has the following objectives in making the compliances of quality implementations.

Objectives

The objective of Centre for Internal Quality Assurance is to develop and put in place a comprehensive and dynamic internal quality assurance system to ensure that programmes of higher education in the Online Learning mode being implemented by the Higher Educational Institution are of acceptable quality and further improved on continuous basis.

Functions of CIQA

The functions of Centre for Internal Quality Assurance would be following:

1. To maintain quality in the services provided to the learners.
2. To undertake self-evaluative and reflective exercises for continual quality improvement in all the systems and processes of the Higher Educational Institution.
3. To contribute in the identification of the key areas in which Higher Educational Institution should maintain quality.
4. To devise mechanism to ensure that the quality of Open and Distance Learning programme and Online programme matches with the quality of relevant programme in conventional mode.
5. To devise mechanisms for interaction with and obtaining feedback from all stakeholders namely, learners, teachers, staff, parents, society, employers, and Government for quality improvement.
6. To suggest measures to the authorities of Higher Educational Institution for qualitative improvement.
7. To facilitate the implementation of its recommendations through periodic reviews.
8. To organize workshops/seminars/symposium on quality related themes, ensure participation of all stakeholders, and disseminate the reports of such activities among all the stakeholders in Higher Educational Institution.

9. To develop and collate best practices in all areas leading to quality enhancement in services to the learners and disseminate the same all concerned in Higher Educational Institution.
10. To collect, collate and disseminate accurate, complete and reliable statistics about the quality of the programme(s).
11. To ensure that Programme Project Report for each programme is according to the norms and guidelines prescribed by the Commission and wherever necessary by the appropriate regulatory authority having control over the programme;
12. To put in place a mechanism to ensure the proper implementation of Programme Project Reports.
13. To maintain a record of Annual Plans and Annual Reports of Higher Educational Institution, review them periodically and generate actionable reports.
14. To provide inputs to the Higher Educational Institution for restructuring of programmes in order to make them relevant to the job market.
15. To facilitate system-based research on ways of creating learner centric environment and to bring about qualitative change in the entire system.
16. To act as a nodal coordinating unit for seeking assessment and accreditation from a designated body for accreditation such as NAAC etc.
17. To adopt measures to ensure internalization and institutionalization of quality enhancement practices through periodic accreditation and audit.
18. To coordinate between Higher Educational Institution and the Commission for various qualities related initiatives or guidelines.
19. To obtain information from other Higher Educational Institutions on various quality benchmarks or parameters and best practices.
20. To record activities undertaken on quality assurance in the form of an annual report of Centre for Internal Quality Assurance.
21. It will be mandatory for Centre for Internal Quality Assurance to submit Annual Reports to the Statutory Authorities or Bodies of the Higher Educational Institution about its activities at the end of each academic session. A copy of report in the format specified by the Commission, duly approved by the statutory authorities of the Higher Educational Institution shall be submitted annually to the Commission.

On enrolling in the BCA Programme at Usha Martin University in Online Learning (OL) mode, students embark on a transformative educational journey that emphasizes critical thinking, effective communication, and advanced problem-solving skills as emphasized in NEP-2020. Through a flexible and learner-centric approach, the programme nurtures a scientific temperament, empowering students to approach challenges with analytical precision and innovative solutions. The curriculum has been designed to instill a strong sense of ethics and responsibility, fostering a mindset that prioritizes sustainability and environmental consciousness. Students are encouraged to engage with real-world problems and develop solutions that contribute positively to society and the global ecosystem.

As they progress, learners gain proficiency in foundational and advanced aspects of computer applications, software development, and emerging technologies. The programme equips them with practical knowledge and skills that align with current industry standards, preparing them for the rapidly evolving landscape of technology. Once the students graduate, he/she will possess the confidence and capability to participate in multiple functional areas of science and technology. They will be well-prepared for dynamic careers in software development, IT services, database management, and more recent developments. Additionally, the programme lays a robust foundation for those aspiring to pursue advanced education or competitive examinations in the public and private sectors.

The BCA Programme at Usha Martin University not only prepares students for professional success but also inspires them to become ethical, innovative, and socially responsible contributors to the global technological community. Through this comprehensive approach, graduates emerge as skilled professionals and leaders ready to shape the future of the digital world.



Registrar
कुल सचिव
Usha Martin University
उषा मार्टिन विश्वविद्यालय
Ranchi-835103 / राँची-835103