MATA TRIPURA SUNDARI OPEN UNIVERSITY, TRIPURA

MTS

MATA TRIPURA SUNDARI
OPEN UNIVERSITY
TRIPURA

PROGRAMME PROJECT REPORT
BACHELOR OF COMPUTER APPLICATION (BCA)
2025–26

Registrar
Mata Tripura Sundari Open University
Gomati-Tripura

Introduction

The Bachelor of Computer Applications (BCA) programme is a dynamic 4-year undergraduate degree designed to provide students with a comprehensive understanding of computer applications and software development fundamentals. Widely regarded as equivalent to a BCA offers a robust academic foundation for those aspiring to excel in the realm of Computer Applications With a targeted curriculum aimed at both advanced education and competitive examinations, BCA offers students a structured path towards higher learning opportunities.

Delivered through a blend of distance learning and ICT-enabled study, the BCA programme mandates a minimum of 160 credits over the course of four years. This flexible learning format accommodates the diverse needs of modern learners while ensuring academic rigor and excellence.

The increasing demand for BCA stems from its potential to unlock various career pathways, including lucrative opportunities in the government sector through competitive examinations, as well as paving the way for fulfilling careers in academia and beyond.

A. Programme's Mission and Objectives

Mission:

The BCA Programme is dedicated to:

- To cater and ensure excellent theoretical and practical training through teaching, counseling, and mentoring with a view to achieve professional and academic excellence.
- To connect with industry and incorporating knowledge for research enhancement.
- To generate, disseminate and preserve knowledge for the benefit and betterment of society.

Objectives:

- To educate and train individuals to be well prepared for higher education.
- To be able to engage independent and life-long learning.
- To develop professionally that ensures existence in the competitive world.

B. Relevance of the Programme with Vision and Goals

The vision and mission of MATA TRIPURA SUNDARI OPEN UNIVERSITY, of Tripura are:

Vision:

The vision for the BCA programme in our institution is to provide a transformative educational experience that empowers students to excel in the dynamic field of computer science and information technology. Through a rigorous curriculum blending theory and practical application, the program aims to equip students with the skills, knowledge, and mindset needed to thrive in today's digital landscape. Emphasizing innovation, collaboration, and diversity, the program fosters a culture of excellence and prepares students to become leaders, innovators, and change makers in the IT industry and beyond. By embracing lifelong learning and continuous improvement, the program remains responsive to the evolving needs of students, industry, and society, ensuring that graduates are well-prepared to make a positive impact in the world through technology.

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 program can contribute to the HEI's goals:

- **Academic Excellence:** The BCA programme can enhance the reputation of the institution by delivering high-quality education in computer science and information technology, attracting talented students and faculty, and producing graduates who excel in their careers.
- **Research and Innovation:** The program can contribute to the institution's research goals by fostering a culture of innovation and inquiry among students and faculty, engaging in cutting-edge research projects, and collaborating with industry partners to develop innovative solutions to real-world problems.
- **Community Engagement:** Through service-learning initiatives, collaborative projects, and outreach programs, the BCA programme can strengthen the institution's ties to the community, demonstrating its commitment to social responsibility and civic engagement.
- **Global Perspective:** By exposing students to global perspectives, multicultural environments, and international collaborations, the program can help the institution fulfill its goal of preparing students to thrive in an increasingly interconnected world.
- **Employability and Career Development:** The program can support the institution's goal of enhancing graduates' employability and career prospects by providing them with the skills, knowledge, and experience needed to succeed in the workforce and adapt to changing industry demands.
- **Diversity and Inclusion:** By promoting diversity, equity, and inclusion within the program, the institution can create a more welcoming and inclusive campus environment, reflecting its commitment to social justice and equality.
- **Continuous Improvement:** Through ongoing assessment, evaluation, and feedback mechanisms, the program can contribute to the institution's goal of continuous improvement, ensuring that it remains responsive to the evolving needs of students, industry, and society.

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 MATA TRIPURA SUNDARI OPEN UNIVERSITY (MTSOU) 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 Application (BCA) programme shall have to meet the eligibility norms as follows –

- To obtain admission in BCA programme, the learner must have completed 10+2 in science stream or equivalent course.
- The BCA programme offered by MATA TRIPURA SUNDARI OPEN 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 to acquire specific skills and competence

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

Programme Outcomes (PO's)

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

- **PO1 (Critical Thinking):** Develop proficiency of techniques, knowledge of vocabulary, to show creative, critical and philosophical thinking of work.
- **PO2 (Professional Skill):** Develop understanding of professional needs, responsibilities, and requirements as an art professional.
- **PO3 (Decision Making):** Show potential in providing creative solutions to communication of complex phenomena of print media such as books, magazines and newspaper, able to depict and perform in group settings.
- **PO4 (Ethics):** Recognize different value systems including their own, understand the moral dimensions of their decisions, and accept responsibility for them.
- **PO5 (Environment and Sustainability):** Understand the issues of environmental contexts and sustainable development.
- **PO6 (Self-directed and Life-long Learning):** Acquire the ability to engage in independent and life-long learning.

Programme Specific Outcomes (PSOs)

PSO1: Enrich the knowledge in the areas like Artificial Intelligence, Web Services, Cloud Computing, Paradigm of Programming language, Design and Analysis of Algorithms, Database Technologies Advanced Operating System, Mobile Technologies, Software Project Management and core computing subjects. Choose to study any one subject among recent trends in IT provided in the optional subjects.
PSO2: Students understand all dimensions of the concepts of software application and projects.
PSO3: Students understand the computer subjects with demonstration of all programming and theoretical concepts with the use of ICT.
PSO4: Developed in-house applications in terms of projects.

E. Instructional Design

The BCA programme is structured into four years, with a minimum credit requirement of 160 to obtain the degree. In MATA TRIPURA SUNDARI OPEN UNIVERSITY, the minimum time period for completing the BCA degree is four years, while the maximum allowable time period is eight years.

Bachelor of Computer Application (BCA)
Course structure and Syllabi
(2024-25)
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 (Hons) 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 Application (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.	CSB-1111	Management Information System	Major	4	30	70	100
2.	CSB-1112	Problem solving Using ‘C’	Major	2	30	70	100
3.	CSB-1113		Minor	4	30	70	100
4.	CSB-1114		Minor	4	30	70	100
5.	ENB-1101	English Communication	AEC	2	30	70	100
6.	CSB-1101	Computer Fundamental & Office Automation	SEC	2	30	70	100
7.	CSB-1151	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.

List of Minors	
CSB-1113	Mathematics-I
CSB-1114	Financial Accounting

Semester – II

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CSB-1211	System Analysis & Design	Major	4	30	70	100
2.	CSB-1212	C++ & Data Structure	Major	2	30	70	100
3.			Minor	4	30	70	100
4.			Minor	4	30	70	100
5.	ENB-1201	Creative Writing	AEC	2	30	70	100
6.	SEC-1201	Reasoning	SEC	2	30	70	100
7.	CSB-1251	C++ & Data Structure Lab	Major	2	30	70	100
Total				**20**	**210**	**490**	**700**

List of Minors	
CSB-1213	Ordinary Differential Equations
CSB-1214	Fundamentals of Electronics

Semester – III

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CSB-2111	Computer Organization	Major	4	30	70	100
2.	CSB-2112	Operating System	Major	2	30	70	100
3.			Minor	4	30	70	100
4.	ENB-2101	Personality Development	AEC	2	30	70	100
5.	VAC-2101	Environmental Education	VAC	3	30	70	100
6.	CSB-2114	Computer Based Numerical Methods	SEC	3	30	70	100
7.	CSB-2151	Operating System Lab	Major	2	30	70	100
Total				**20**	**210**	**490**	**700**

List of Minors	
CSB-2113	Integral Calculus

Semester – IV

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CSB-2211	Analysis and Design of Algorithms	Major	4	30	70	100
2.	CSB-2212	Object Oriented Programming with Java	Major	2	30	70	100
3			Minor	4	30	70	100
4.	CSB-2214	Database Management System	SEC	3	30	70	100
5.	ENB-2201	Basic Knowledge of English	AEC	2	30	70	100
6.	VAC-2201	Understanding India	VAC	3	30	70	100
7.	CSB-2251	Object Oriented Programming with Java Lab	Major	2	30	70	100
Total				**20**	**210**	**490**	**700**

List of Minors	
CSB-2213	Matrix Theory

Semester – V

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1.	CSB-3111	Software Engineering	Major	4	30	70	100
2.	CSB-3112	Computer Networks	Major	4	30	70	100
3.	CSB-3113	Theory of Computation	Major	4	30	70	100
4.	CSB-3114	Perspective of Physics	Interdisciplinary	3	30	70	100
5.	CSB-3115	Basic Abstract & Linear Algebra	Interdisciplinary	3	30	70	100
6.	CSB-3151	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	CSB-3211	IoT & Cloud Computing	Major	4	30	70	100
2	CSB-3212	Machine Learning Using Python	Major	4	30	70	100
3	CSB-3213	Discrete Mathematics	Major	4	30	70	100
4	CSB-3214	Concepts of Physics	Interdisciplinary	3	30	70	100
5	CSB-3251	Machine Learning Using Python Lab	Major	2	30	70	100
6	CSB-3291	Internship	SEC	3	30	70	100
Total				**20**	**180**	**420**	**600**

Semester – VII

S.N.	Course Code	Course Name	Category	Credits	Continuous Assessment Marks	Term End Exam Marks	Grand Total
1	CSB-4111	Soft Computing	Major	4	30	70	100
2	CSB-4112	R Programming	Major	4	30	70	100
3	CSB-4113	Network Security & Cryptography	Major	4	30	70	100
4.	CSB-4114	Parallel Computing	Major	4	30	70	100
5.	CSB-4115	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.	CSB-4211	Natural Language Processing	Major	4	30	70	100
2.	CSB-4212	Web Designing	Major	4	30	70	100
3.	CSB-4213	Data Warehousing & Data Mining	Major	4	30	70	100
4.	CSB-4214	Digital Forensics	Major	4	30	70	100
5.	CSB-4215	Bioinformatics using AI	Minor	4	30	70	100
Total				**20**	**150**	**350**	**500**

SEMESTER – I

Course Code: CSB–1111 **Credit: 4**
Course Name: Management Information System

Course Objectives:

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

Course Outcomes:

By the end of the course, students should be able to:

1. Demonstrate a solid understanding of fundamental concepts and theories related to Management Information Systems, including the role of information systems in organizations, the components of information technology infrastructure, and the principles of database management.
2. Analyze business requirements and organizational needs to design and develop effective information systems solutions, including database design, system architecture, and user interface design, aligning IT initiatives with strategic business objectives.
3. Utilize information systems tools and technologies effectively to collect, process, store, and disseminate information within organizations, enhancing decision-making processes, operational efficiency, and collaboration among stakeholders.
4. Evaluate the performance and effectiveness of information systems in meeting organizational objectives, using key performance indicators (KPIs) and metrics to assess system reliability, security, scalability, and usability.
5. Apply MIS principles strategically to address contemporary business challenges, such as digital transformation, cybersecurity threats, data analytics, and emerging technologies, fostering innovation and driving organizational growth and sustainability.

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 case 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: Case Studies and Future Directions in MIS: Case studies of MIS in various industries, Future directions and trends in MIS, Integrative project on MIS strategy

References:

1. Laudon, K. C., & Laudon, J. P. (2020). Management Information Systems: Managing the Digital Firm (16th ed.). Pearson.
2. Oz, E. (2019). Management Information Systems (8th ed.). Cengage Learning.
3. Stair, R. M., & Reynolds, G. W. (2020). Principles of Information Systems (13th ed.). Cengage Learning.
4. Turban, E., &Volonino, L. (2019). Information Technology for Management: Advancing Sustainable, Profitable Business Growth (11th ed.). Wiley.
5. Kroenke, D. M., & Boyle, R. J. (2020). Using MIS (11th ed.). Pearson.
6. Baltzan, P. (2019). Business Driven Technology (7th ed.). McGraw-Hill Education.

Course Code: CSB–1112 **Credit: 2**

Course Name: Problem solving Using 'C'

Course Objective:

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

Course Outcomes:

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

1. Students will demonstrate proficiency in fundamental programming concepts, including variables, control structures, functions, and data types, enabling them to write basic programs in a variety of programming languages.
2. Students will develop the ability to analyze problems, design algorithms, and implement solutions using appropriate programming constructs. They will learn various problem-solving strategies and techniques for algorithmic design and analysis.
3. Through hands-on coding exercises, students will learn to write structured and efficient code following best practices in software development. They will understand the importance of code readability, reusability, and maintainability.
4. Students will apply algorithmic concepts and data structures to solve computational problems across different domains, including sorting, searching, graph traversal, and dynamic programming.
5. This course will foster critical thinking and logical reasoning skills as students tackle algorithmic challenges and analyze the efficiency of algorithms. They will learn to evaluate the correctness and efficiency of their solutions and make informed decisions based on computational constraints.

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. Cormen, Leiserson, Rivest, Stein, “Introduction to Algorithms", PHI.
2. Basse, "Computer Algorithms: Introduction to Design & Analysis", Addision Wesley.
3. Horowitz &Sahani, "Fundamental of Computer Algorithm", Galgotia.
4. Dasgupta, Papadimitriou &Vazirani, "Algorithms". TMH. (An earlier version is freely available online)

Course Code: CSB-1113 **Credits: 4**
Course Name: Mathematics-I

Course Objectives:

- Provide students with a foundational understanding of essential mathematical concepts.
- Ensure comprehension of sets and functions, properties of numbers, matrices, and systems of linear equations.
- Teach key calculus concepts, including limits, continuity, differentiability, and integration.
- Prepare students for further studies in mathematics and related fields.
- Enable students to analyse problems and apply mathematical techniques with confidence.

Course Outcomes: After completing this course, students will be able to

1. Define set, relation and functions
2. Understand and apply the principle of mathematical induction
3. Do various operations on matrices
4. Investigate the continuity and differentiability of functions
5. Evaluate various types of integrals
6. Analyze the consistency and inconsistency of system of linear equations

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:

- H. Anton, I. Birens and S. Davis, Calculus, John Wiley and Sons, Inc.,2002.
- G.B. Thomas and R.L. Finney, Calculus, Pearson Education,2007.
- T. M. Apostol, Calculus Vol I, Wiley & Sons (Asia) Pvt.Ltd.
- Gorakh Prasad, Differential Calculus, PothishalaPvt. Ltd.,Allahabad

Course Code: CSB-1114 **Credits: 4**
Course Name: Financial Accounting

Course Objective:

- Provide students with a comprehensive understanding of the principles and concepts of financial accounting.
- Teach the methods of recording, classifying, and summarizing financial transactions.
- Develop the ability to prepare and interpret financial statements, including balance sheets, income statements, and cash flow statements.
- Enhance students' skills in analyzing financial data to make informed business decisions.
- Ensure students understand the regulatory environment and ethical considerations in financial accounting.

Course Outcomes:
After completing the course, the student shall be able to:

1. Enhanced financial decision-making through the analysis and interpretation of financial statements.
2. Comprehensive understanding of compliance and regulatory requirements in financial reporting.
3. Improved skills in accurately recording, summarizing, and reporting financial transactions.
4. Ability to create and manage budgets, forecast financial performance, and plan strategically.
5. Proficiency in auditing principles and internal controls to ensure financial data integrity and fraud prevention.

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, Cash 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. Goyal, Bhushan Kumar and H.N. Tiwari, Financial Accounting, Taxmann
2. Kumar, Alok. Financial Accounting, Singhal Publication.
3. Lt Bhupinder. Financial Accounting – Concepts and Applications, Cengage
4. Monga, J R. Financial Accounting: concept and Applications. Mayur paper Backs, New Delhi.

Course Code: ENB–1101 **Credit: 2**
Course Name: English Communication

Course Objective:

- Equip students with essential skills and confidence for effective English communication.
- Develop proficiency in speaking, listening, reading, and writing.
- Enhance interpersonal communication abilities in various contexts.
- Improve professional communication skills.
- Provide a comprehensive curriculum to support language development.

Course Outcomes:

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

1. Students will demonstrate the ability to express themselves clearly and confidently in spoken English, including participating in discussions, delivering presentations, and engaging in interpersonal communication.
2. Students will develop active listening skills, enabling them to comprehend and respond appropriately to spoken English in various situations, including lectures, conversations, and presentations.
3. Students will enhance their reading comprehension skills by analyzing and interpreting various forms of written English, including articles, essays, and literary texts, with an emphasis on vocabulary expansion and understanding context.
4. Students will learn to write clearly and coherently in English, producing well-structured essays, reports, emails, and other written documents suitable for academic, professional, and personal communication.
5. Students will develop an understanding of cultural nuances and conventions in English communication, fostering intercultural sensitivity and adaptability in diverse linguistic and cultural contexts.

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. (2019). "Business Communication: Process and Product." Cengage Learning.

2. Lesikar, R. V., Flatley, M. E., &Rentz, K. (2017). "Basic Business Communication: Skills for Empowering the Internet Generation." McGraw-Hill Education.

3. O'Hair, D., Wiemann, M., & Mullin, D. I. (2018). "Real Communication: An Introduction." Bedford/St. Martin's.

4. Adler, R. B., & Elmhorst, J. M. (2016). "Communicating at Work: Principles and Practices for Business and the Professions." McGraw-Hill Education.

5. Hargie, O. (2019). "Skilled Interpersonal Communication: Research, Theory and Practice." Routledge.

Course Code: CSB–1101 **Credit: 2**

Course Name: Computer Fundamental & Office Automation

Course Objective:

- Introduce students to basic computer concepts and operating systems.
- Teach the fundamental principles of hardware and software components.
- Develop proficiency in using office automation tools such as word processors, spreadsheets, and presentation software.
- Enhance understanding of computer networks and internet applications.
- Equip students with practical skills for efficiently managing and automating office tasks.

Course Outcomes:

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

1. Foundational Knowledge: Students will demonstrate a solid understanding of computer fundamentals, including hardware components, operating systems, and software applications.
2. Proficiency in Office Automation: Students will acquire proficiency in using office automation tools such as word processing, spreadsheets, presentations, and email communication software.
3. Efficient Office Practices: Students will develop the ability to apply office automation tools effectively to enhance productivity and efficiency in office tasks, including document creation, data analysis, and communication.
4. Problem-Solving Skills: Through practical exercises and real-world applications, students will enhance their problem-solving skills, enabling them to troubleshoot common computer and software issues encountered in office environments.
5. Effective Communication: Students will learn to communicate effectively using office automation tools, producing clear and professional documents, presentations, and email correspondence suitable for various business contexts.

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

Course Code: CSB–1151 **Credit: 2**
Course Name: Problem solving Using 'C' Lab

Course Objective:

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

Course Outcomes:

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

1. Students will gain practical experience in applying programming concepts learned in the classroom to solve real-world problems. They will develop the ability to translate algorithmic solutions into working code using a chosen programming language.
2. Students will enhance their coding skills and become proficient in writing clean, efficient code. They will also learn debugging techniques to identify and fix errors in their programs effectively.
3. Students will implement algorithms discussed in theoretical courses, such as sorting, searching, and data structure manipulation, through hands-on programming assignments. They will analyze the efficiency and effectiveness of different algorithmic approaches.
4. Through individual and group projects, students will have the opportunity to apply programming and algorithmic concepts to solve complex problems. They will learn to manage project requirements, collaborate effectively with peers, and present their solutions to the class.
5. The lab exercises and projects will foster critical thinking and problem-solving skills as students tackle algorithmic challenges and optimize their code for efficiency and performance. They will learn to evaluate the trade-offs between different solutions and make informed decisions based on computational constraints.

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

Semester – II

Course Code: CSB–1211 **Credit: 4**

Course Name: System Analysis & Design

Course Objective:

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

Course Outcomes:

Upon successful completion of the course, students will be able to:

1. Understand the life cycle of a systems development project.
2. Understand the analysis and development techniques required as a team member of a medium-scale information systems development project
3. Learn the effect of internet and technology on business strategies.
4. Understand the importance of business communications.
5. An understanding of the object-oriented methods models as covered by the Unified Modelling Language.

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, Behaviorassues.

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. Elias M. Awad, "Systems Analysis & Design", Galgotia Publication.

2. Hoffer, "Modern Systems Analysis & Design", Addision Wesley.

3. Kendall, "Introduction to System Analysis & Design", McGraw Hill

Course Code: CSB–1212 **Credit: 2**
Course Name: C++ & Data Structure

Course Objective:

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

Course Outcomes:

Upon successful completion of the course, students will be able to:

1. Students will demonstrate proficiency in implementing various data structures such as arrays, linked lists, stacks, queues, trees, and graphs using C++, understanding their properties, operations, and applications.
2. Students will be able to analyze the time and space complexities of algorithms, allowing them to make informed decisions about the suitability of different data structures for specific problem-solving scenarios.
3. Students will develop problem-solving skills through hands-on coding exercises and programming assignments, enabling them to translate abstract concepts into practical solutions.
4. Students will gain proficiency in algorithmic techniques such as sorting, searching, recursion, and dynamic programming, facilitating the efficient manipulation and processing of data.
5. Students will enhance their software engineering skills by applying best practices in code design, documentation, and optimization, fostering a systematic approach to software development and maintenance.

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-case
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. Thareja, “Data Structure Using C” Oxford Higher Education.
2. AK Sharma, “Data Structure Using C”, Pearson Education India.
3. Rajesh K. Shukla, “Data Structure Using C and C++” Wiley Dreamtech Publication.
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount “Data Structures and Algorithms in C++”, WileyIndia.

Course Code: CSB-1213 **Credits: 4**

Course Name: Ordinary Differential Equations

Course Objectives:

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

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

1. Define linear differential equations with constantcoefficient.
2. Solve first order differential equations including separable, homogeneous, exact and linear.
3. Calculate complementary function (C.F.) and particular integral(P.I.).
4. Find Series solutions of second order differential equations.
5. Understand Legendre and Bessel functions (Pn and Jn only) and their properties

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 Bernaulli'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 (Pn and Jn only) and their properties.

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

References:

- Barnes, Belinda & Fulford, Glenn R. (2015). Mathematical Modeling with Case Studies, Using Maple and MATLAB (3rd ed.). CRC Press, Taylor & Francis Group.
- Edwards, C. Henry, Penney, David E., & Calvis, David T. (2015). Differential Equation and Boundary Value Problems: Computing and Modeling (5th ed.).Pearson Education.

- Ross, Shepley L. (2004). Differential Equations (3rd ed.). John Wiley &Sons. India.

Course Code: CSB-1214 **Credits: 4**

Course Name: Fundamentals of Electronics

Course Objectives:

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

Course Outcomes:

1. Demonstrate a clear understanding of fundamental electronic concepts, including voltage, current, resistance, and power.
2. Analyze and solve basic electronic circuits using Ohm's Law, Kirchhoff's Laws, and network theorems.
3. Identify and explain the operation of key semiconductor devices such as diodes, BJTs, and FETs.
4. Design and construct simple electronic circuits, including amplifiers, oscillators, and power supplies.
5. Apply knowledge to practical scenarios, effectively troubleshooting and optimizing basic electronic systems.

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. (2009). Electronic devices and circuit theory (11th ed.). Pearson Education.
2. Horowitz, P., & Hill, W. (2015). The art of electronics (3rd ed.). Cambridge University Press.
3. Scherz, P., & Monk, S. (2016). Practical electronics for inventors (4th ed.). McGraw-Hill Education.
4. Alexander, C. K., &Sadiku, M. N. O. (2016). Fundamentals of electric circuits (6th ed.). McGraw-Hill Education.
5. Floyd, T. L. (2017). Electronic devices (Electron flow version) (10th ed.). Pearson Education.

Course code: ENB-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:

1. Students will be able to generate original ideas and develop them into well-crafted pieces of writing.
2. Students will demonstrate proficiency in different writing styles and techniques.
3. Students will be able to effectively communicate their thoughts and emotions through their writing.
4. Students will have a solid understanding of various forms of creative writing and their unique characteristics.
5. Students will be able to give and receive constructive feedback on their own and others' work.

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. "Creating Characters: How to Build Story People" by Dwight V. Swain.
2. "Self-Editing for Fiction Writers: How to Edit Yourself into Print" by Renni Browne and Dave King.
3. "The Creative Writing Coursebook: Forty Authors Share Advice and Exercises for Fiction and Poetry" edited by Julia Bell and Paul Magrs.
4. "The Making of a Story: A Norton Guide to Creative Writing" by Alice LaPlante.
5. "The Power of Point of View: Make Your Story Come to Life" by Alicia Rasley.

Code: SEC–1201 **Credit: 2**
Course Name: Reasoning

Course Objectives:

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

Course Outcomes:
Upon completing the course, learners will be able to:

1. Apply Logical and Analytical Reasoning: Accurately solve a wide range of logical and analytical reasoning questions, using deductive and inductive reasoning skills effectively in both academic and real-world scenarios.
2. Demonstrate Enhanced Critical Thinking: Critically assess arguments, identify logical fallacies, make informed decisions, and construct coherent arguments of their own, applying these skills in diverse contexts.
3. Interpret and Analyze Data Competently: Read and interpret complex data from charts, graphs, and tables, and perform data sufficiency tasks with proficiency, essential for success in the quantitative sections of competitive exams.

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."A Modern Approach to Logical Reasoning" by R.S. Aggarwal
2."Introduction to Logic" by Irving M. Copi, Carl Cohen, and Kenneth McMahon
3."How to Think Logically" by Gary Seay and Susana Nuccetelli

Course Code: CSB–1251 **Credit: 2**
Course Name: C++ & Data Structure Lab

Course Objective:

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

Course Outcomes:

Upon successful completion of the 'C' Programming Lab, students will be able to:

1. Understand the concept of data structures, python and apply algorithm for solving problems like Sorting, searching, insertion and deletion of data.
2. Understand linear data structures for processing of ordered or unordered data.
3. Ability to understand a systematic approach to organizing, writing and debugging C programs
4. Ability to implement linear and non-linear data structure operations using C programs
5. Ability to solve problems implementing appropriate data structures
6. Ability to implement sorting and searching algorithms using relevant data structures

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.

Semester - III

Course Code: CSB-2111 **Credit: 4**

Course Name: Computer Organization

Course Objective:

Computer Organization and Architecture is a comprehensive course designed to provide students with a deep understanding of the structure and functionality of computer systems. This course covers topics ranging from basic digital logic design to the architecture of modern computer systems. Through lectures, laboratory sessions, and practical assignments, students will gain insights into the organization of computer hardware, instruction set architecture, memory systems, and input/output mechanisms.

Course Outcomes:

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

1. Understanding of Basic Computer Architecture Concepts.
2. Proficiency in Assembly Language Programming
3. Comprehension of Processor Design and Performance Evaluation.
4. Knowledge of Memory Systems and Storage Technologies.
5. Awareness of Parallel and Distributed Computing Concepts.

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. Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy
2. Structured Computer Organization" by Andrew S. Tanenbaum.
3. Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson
4. Computer Systems: A Programmer's Perspective" by Randal E. Bryant and David R. O'Hallaron
5. Modern Processor Design: Fundamentals of Superscalar Processors" by John Paul Shen and Mikko H. Lipasti.

Course Code: CSB–2112 **Credit:2**
Course Name: Operating System

Course Objective:

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

Course Outcomes:

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

1. Understand the architecture and components of operating systems, including the kernel, device drivers, and system libraries.
2. Analyze and compare different types of operating systems, such as batch processing systems, time-sharing systems, and real-time systems, based on their design and functionality.
3. Demonstrate proficiency in process management, including process creation, scheduling, synchronization, and inter-process communication.
4. Implement and manage memory management techniques, including memory allocation, segmentation, paging, and virtual memory, to optimize system performance and resource utilization.
5. Design and implement file systems, including file organization, directory structures, and file access methods, to manage storage resources efficiently.

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), Case 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 TechniquesFixed 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, Galvin and Gagne, “Operating Systems Concepts”, Wiley
2. Sibsankar Halder and Alex A Aravind, “Operating Systems”, Pearson Education
3. Harvey M Dietel, “ An Introduction to Operating System”, Pearson Education
4. D M Dhamdhere, “Operating Systems : A Concept based Approach”, 2nd Edition,TMH
5. William Stallings, “Operating Systems: Internals and Design Principles ”, 6th Edition,Pearson Ed

Course Code: CSB-2113 **Credits:4**
Course Name: Integral Calculus

Course Objectives:

- Provide a comprehensive understanding of integral calculus concepts, including antiderivatives, definite integrals, and the fundamental theorem of calculus.
- Teach various integration techniques, such as substitution, integration by parts, and trigonometric substitution.
- Develop proficiency in calculating areas, volumes, and other geometric quantities using integration.
- Explore applications of integral calculus in physics, engineering, economics, and other disciplines.
- 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 shall be able to

1. Understand the concept of a partition and Riemannintegration.
2. Differentiate between Improperintegrals.
3. Evaluate ReductionFormulae.
4. Apply integration techniques to find volume andsurfaces.
5. Analyze and apply the concept of vector differentiation andintegration.

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:

- H. Anton, I. Birens and S. Davis, Calculus, John Wiley and Sons, Inc.,2002.

- G.B. Thomas and R.L. Finney, Calculus, Pearson Education,2007.
- T. M. Apostol, Calculus Vol I, Wiley & Sons (Asia) Pvt.Ltd.
- Gorakh Prasad, Differential Calculus, PothishalaPvt. Ltd.,Allahabad

Course code: ENB-2101 **Credit: 2**
Course title: Personality Development

Course Objectives:

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

Course Outcomes:

Upon successful completion of this course, participants will be able to:

- Demonstrate a deeper understanding of their personality, including strengths, limitations, and potential areas for growth.
- Employ improved communication skills, adapting their approach to suit various audiences and contexts.
- Apply emotional intelligence in personal and professional relationships to foster understanding and cooperation.
- Navigate various situations confidently, making decisions assertively and presenting ideas persuasively.

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., and Barry Z. Posner. "The Leadership Challenge: How to Make Extraordinary Things Happen in Organizations." Jossey-Bass, 2017.

Course code: VAC-2101

Course Title: Environmental Education **Credit: 3**

Course Objectives:

- Students will 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.
- Students will be introduced to global, regional, and local environmental challenges, including pollution, resource depletion, and biodiversity loss, understanding their causes and effects.
- The course aims to equip students with the knowledge and skills to develop and evaluate sustainable solutions to environmental challenges, emphasizing the role of innovation and technology.
- Students will be encouraged to reflect on their roles and responsibilities in mitigating environmental issues, promoting a sense of stewardship towards the planet.

Course Outcomes:
Upon successful completion of this course, students will be able to:

1. Demonstrate a comprehensive understanding of environmental systems and the interdependenciesbetween humans and the natural world.
2. Identify key environmental challenges and critically assess their causes, impacts, and the complexities involved in addressing them.
3. Apply knowledge of environmental science and sustainable practices to develop, propose, and evaluate solutions to environmental problems.
4. Exhibit a commitment to environmental responsibility in personal and professional contexts, including sustainable lifestyle choices and advocacy for environmental causes.

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

Reference:

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.
   McKinney, Michael L., Robert M. Schoch, and Logan Yonavjak. "Environmental Science: Systems and Solutions." Jones & Bartlett Learning, 5th edition, 2012.

Course Code: CSB-2114 **Credit: 3**

Course Name: Computer Based Numerical Analysis

Course Objectives:

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

Course Outcomes:

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

1. Applyerroranalysistechniquestoquantifyandminimizeinaccuraciesinnumericalcomputations.
2. Employ various methods for finding solutions to algebraic and transcendental equations.
3. Demonstrate proficiency in interpolating functions using different techniques and working with unequal intervals.
4. Utilize numerical differentiation & integration methods to approximate derivatives and integrals.

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

Reference:

1. Ranganatham S., Prasad M. V. S. S. N. and Ramesh Babu V.; Numerical Analysis, S. Chand Publishing.
2. Jain M. K., Iyengar S.R.K., Jain R. K.; Numerical Methods, New AgeInternational
3. Sastry S. S.; Introductory Methods of Numerical Analysis, PHI Learning Pvt.Ltd.
4. Hildebrand F. B.; Introduction to Numerical Analysis, McGraw-HillEducation.
5. Gupta R. S.; Elements of Numerical Analysis, Algebra, Cambridge UniversityPress.
6. Conte S. and Deboor C.; Elementary Numerical Analysis, McGraw-HillEducation.

Course Code: CSB-2151 **Credit: 2**
Course Name: Operating System Lab

Course Objective:

- Provide practical experience in working with operating system concepts and functionalities.
- Familiarize students with various operating system commands, utilities, and tools.
- Develop proficiency in system administration tasks such as user management, file system operations, and process management.
- Offer hands-on exercises to reinforce theoretical concepts learned in the operating systems course.
- Prepare students for troubleshooting common operating system issues and challenges.

Course Outcomes:

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

1. Gain practical experience in implementing operating system concepts such as process scheduling, synchronization, and deadlock handling.
2. Develop skills in system call programming and process management techniques, including creation, termination, and communication between processes.
3. Understand memory management principles, including allocation, deallocation, and memory protection mechanisms, through hands-on exercises and simulations.
4. Master file system operations and disk management techniques, including file creation, deletion, and manipulation, enhancing proficiency in storage management within an operating system environment.

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. "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne
2. "Operating Systems: Internals and Design Principles" by William Stallings
3. "Modern Operating Systems" by Andrew S. Tanenbaum, Herbert Bos
4. "Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture" by Richard McDougall, Jim Mauro

Semester - IV

Course Code: CSB-2211 **Credit: 4**
Course Name: Analysis and Design of Algorithm

Course Objective:

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

Course Outcomes:

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

1. Students will develop the ability to formulate and solve algorithmic problems using various design paradigms, including divide and conquer, dynamic programming, and greedy algorithms.
2. Students will gain a deep understanding of algorithmic complexity analysis techniques, including time complexity, space complexity, and asymptotic notation, enabling them to evaluate the efficiency of algorithms rigorously.
3. Students will acquire proficiency in implementing algorithms in a programming language of their choice, translating theoretical concepts into practical code and gaining hands-on experience with algorithmic implementation.
4. Through problem-solving exercises and algorithmic analysis, students will develop critical thinking and analytical skills, enabling them to approach complex problems systematically and devise efficient algorithmic solutions.

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-case, average-case, and best-case 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." MIT Press, 2009.
2. Kleinberg, Jon, and ÉvaTardos. "Algorithm Design." Pearson, 2005.
3. Dasgupta, Sanjoy, Christos Papadimitriou, and Umesh Vazirani. "Algorithms." McGraw-Hill Education, 2008.
4. Sedgewick, Robert, and Kevin Wayne. "Algorithms." Addison-Wesley Professional, 2011.
5. Skiena, Steven S. "The Algorithm Design Manual." Springer, 2008.

Course Code: CSB-2212 **Credit: 2**

Course Name: Object Oriented Programming with JAVA

Course Objective:

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

Course Outcomes:

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

1. Students will gain proficiency in writing Java code to implement object-oriented concepts, including classes, objects, methods, and constructors, to solve computational problems.
2. Students will understand and apply key object-oriented design principles such as encapsulation, inheritance, and polymorphism to design robust and flexible software solutions.
3. Students will develop software development skills, including modular design, code organization, and debugging techniques, enabling them to create well-structured and maintainable Java applications.
4. Students will be able to design, develop, and deploy Java-based applications, including graphical user interfaces (GUIs), data processing applications, and web applications, demonstrating their ability to apply OOP concepts to real-world software development scenarios.

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." Pearson, 2018.
2. Savitch, Walter. "Java: An Introduction to Problem Solving and Programming." Pearson, 2018.
3. Eckel, Bruce. "Thinking in Java." Prentice Hall, 2006.
4. Horstmann, Cay S. "Core Java Volume I--Fundamentals." Prentice Hall, 2016.
5. Bloch, Joshua. "Effective Java." Addison-Wesley Professional, 2017.

Course Code: CSB-2213 **Credits: 4**
Course Name: Matrix Theory

Course Objectives:

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

Course Outcomes:

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

6. Understand the concept of amatrix.
7. Do basic operation on matrices like addition andmultiplication.
8. Calculate the determinant, rank of a matrix, echelon and normal form of amatrix.
9. Decide whether a system of linear equations is consistent ofinconsistent.
10. Determine the eigenvalues and eigenvectors of a squarematrix
11. Understand, analyze and classify the quadratic and bilinearforms

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. Springer Science & Business Media, 2011.
2. Abadir, Karim M., and Jan R. Magnus. Matrix algebra. Vol.1. Cambridge University Press, 2005.
3. Narayan, Shanti, and P. K. Mittal. A text bookofmatrices.S.ChandPublishing,2010.
4. Singh, Arindama. Introduction to matrix theory. Springer,2021.
5. Datta, Kanti Bhushan. Matrix and Linear Algebra aided with MATLAB. PHI Learning Pvt. Ltd., 2016.
6. Serre, Denis, and Denis Serre. What are matrices. Springer New York,2010.

Course Code: CSB-2214 **Credit: 3**
Course Name: Database Management System

Course Objective:

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

Course Outcomes:

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

1. Understand the fundamental concepts and principles of Database Management Systems (DBMS).
2. Design and implement relational databases using appropriate data modeling techniques.
3. Utilize Structured Query Language (SQL) to create, retrieve, update, and delete data from databases.
4. Implement normalization techniques to ensure data integrity and minimize redundancy in database design.
5. Apply indexing and optimization techniques to enhance database performance and query efficiency.

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, Cascading and Cascade 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. Bipin C. Desai, “An introduction to Database Systems”, Galgotia Publication.
2. Korth, Silbertz, Sudarshan, “Database Concepts”, McGraw Hill.
3. Date C J, “An Introduction To Database System”, Addision Wesley.
4. Maheshwari Jain, “DBMS: Complete Practical Approach”, Firewall Media, New Delhi.
5. ITL education solution Ltd “Introduction to database System ”, Pearson, New Delhi.

Course code: ENB-2201 **Credit: 2**
Course Title: Basic Knowledge of English

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:

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

1. Students will be able to identify and apply key grammar rules, such as subject-verb agreement, tense usage, and sentence structure.
2. Students will be able to effectively use parts of speech, including nouns, pronouns, verbs, adjectives, and adverbs.
3. Students will be able to recognize and correct common grammatical errors in their own writing.
4. Students will be able to demonstrate improved proficiency in grammar through quizzes, exams, and class assignments.
5. Students will be able to communicate clearly and confidently in both informal and formal settings using correct grammar.

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*. Cambridge University Press.
2. O'Conner, Patricia T. *Woe is I: The Grammarphobe's Guide to Better English in Plain English*. Riverhead Books.
3. Strunk Jr., William, and E.B. White. *The Elements of Style*. Pearson.
4. Thurman, Susan. *The Only Grammar Book You'll Ever Need: A One-Stop Source for Every Writing Assignment*. Adams Media.

Course code: VAC-2201 **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:

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

1. Have a deep understanding of the Indian Constitution, its historical context, and its current application.
2. Be knowledgeable about the fundamental rights and duties of Indian citizens and their significance.
3. Understand the roles and functions of various pillars of Indian democracy, including the executive, legislature, and judiciary.
4. Be aware of the socio-political challenges facing India and the measures being taken to address them.
5. Be able to critically analyze contemporary issues in the Indian socio-political context and their constitutional implications.

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: CSB-2251 **Credit: 2**

Course Name: Object Oriented Programming with Java Lab

Course Objective:

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

Course Outcomes:

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

1. **Java Programming Proficiency:** Students will demonstrate competence in Java programming, including syntax, data structures, control flow, and object-oriented concepts such as classes, objects, inheritance, polymorphism, and encapsulation.
2. **Object-Oriented Design Skills:** Students will learn to apply object-oriented design principles such as abstraction, encapsulation, inheritance, and polymorphism to model and solve real-world problems effectively.
3. **Software Development Practices:** Students will gain experience in software development practices such as modular design, code organization, documentation, and testing, fostering habits of writing clean, maintainable, and well-documented code.
4. **Problem-Solving Abilities:** Students will develop problem-solving skills by working on a variety of programming assignments and projects, which require them to analyze requirements, design appropriate solutions, and implement them using object-oriented programming techniques.

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. Cay S. Horstmann, Gary Cornell - "Core Java Volume I--Fundamentals" (11th Edition)
2. Cay S. Horstmann, Gary Cornell - "Core Java Volume I--Fundamentals" (11th Edition)
3. Herbert Schildt - "Java: A Beginner's Guide" (8th Edition)
4. Kathy Sierra, Bert Bates - "Head First Java" (2nd Edition)
5. Cay S. Horstmann, Gary Cornell - "Core Java Volume II--Advanced Features" (11th Edition)
6. Collections Framework: Experiment: Utilize Java Collections Framework to work with data structures such as lists, sets, maps, and queues. Reference: Joshua Bloch - "Effective Java" (3rd Edition)

Semester – V

Course Code: CSB-3111 **Credit: 4**
Course Name: Software Engineering

Course Objective:

- Introduce students to the fundamental principles and concepts of software engineering.
- Teach software development methodologies, including requirements analysis, design, implementation, testing, and maintenance.
- Develop skills in software project management, including planning, scheduling, and tracking.
- Explore software quality assurance techniques, such as testing strategies, code reviews, and software metrics.
- Provide hands-on experience with software development tools and techniques through practical projects and case studies.

Course Outcomes:

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

1. Students will gain a thorough understanding of the software development lifecycle (SDLC) and various software development methodologies, including waterfall, agile, and DevOps, enabling them to choose and apply appropriate methodologies based on project requirements.
2. Students will develop proficiency in gathering, analyzing, and documenting software requirements, translating user needs into functional specifications, and managing requirements throughout the software development process.
3. Students will be able to design software architectures and components using appropriate design patterns, principles, and modeling techniques, and implement software solutions using programming languages and development tools.
4. Students will acquire project management skills, including planning, scheduling, budgeting, and risk management, and understand the importance of teamwork, communication, and collaboration in successful software development projects.

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 case 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 case 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, 2019.
2. Sommerville, Ian. "Software Engineering." Pearson, 2015.
3. Sommerville, Ian. "Software Engineering: International Edition." Pearson, 2015.
4. Pfleeger, Shari Lawrence, and Joanne M. Atlee. "Software Engineering: Theory and Practice." Pearson, 2018.
5. Ian, Sommerville. "Software Engineering (10th Edition)." Pearson, 2015.
6. McConnell, Steve. "Code Complete: A Practical Handbook of Software Construction." Microsoft Press, 2004.

Course Code: CSB-3112 **Credit: 4**
Course Name: Computer Networks

Course Objective:

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

Course Outcomes:

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

1. Understand the fundamental concepts and principles of computer networks, including network architecture, network models, and network protocols.
2. Analyze and compare different types of network topologies, such as bus, star, ring, mesh, and hybrid topologies, based on their advantages and disadvantages.
3. Configure and troubleshoot network devices, such as routers, switches, hubs, and access points, to establish and maintain network connectivity.
4. Implement and analyze various network protocols, including TCP/IP, UDP, HTTP, FTP, SMTP, DHCP, DNS, and SNMP, to facilitate communication between network devices.
5. Design and implement secure networks using encryption, authentication, access control, and intrusion detection mechanisms to protect network resources and data.

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, 4thed, McGraw Hill.
2. A.S. Tanenbaum, “Computer Networks”, 4th Edition, Prentice Hall India, 1997.
3. S. Keshav, “An Engineering Approach on Computer Networking”, Addison Wesley, 1997
4. W. Stallings, “Data and Computer Communication”, Macmillan Press, 1989.

Course Code: CSB-3113 **Credit: 4**
Course Name: Theory of Computation

Course Objective:

- Introduce students to the theoretical foundations of computation and formal languages.
- Teach key concepts such as automata theory, formal grammars, and Turing machines.
- Develop proficiency in analyzing the computational power and limitations of different models of computation.
- Explore topics such as regular languages, context-free languages, and computability theory.
- Provide insight into the relationships between theoretical models of computation and practical applications in computer science.

Course Outcomes:

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

1. Students will be able to explain the theoretical foundations of computer science, including automata theory, formal languages, and computational complexity.
2. Students will be able to analyze and design finite automata, regular expressions, and context-free grammars for solving language-related problems.
3. Students will be able to model computational processes using Turing machines and understand their computational power and limitations.
4. Students will be able to analyze the time and space complexity of algorithms, classify problems into complexity classes, and recognize NP-complete problems.
5. Students will be able to apply theoretical concepts learned in the course, such as finite state machines and formal languages, to solve practical problems in software development and formal language processing.

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
2. K. L. P Mishra, N. Chandra shekaran (2003), Theory of Computer Science-AutomataLanguages and Computation, 2nd edition, Prentice Hall of India, India.

Course Code: CSB – 3114 **Credit: 3**

Course Name: Perspectives in Physics

Course Objective

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

Course Outcomes:

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

1. To know the concepts of Atoms, Nucleus and Nuclear Energy, Relativity, elementary particles and cosmology.
2. To solve problems based on the fundamental concepts.
3. To name quarks and another elementary particles.

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,(2009)
2. IntroductionToPhysicsJohnDCutnell.K.W.Johnson,DYoung,SStadlerWileyIndiaEdition, (2015)
3. Fundamentals Of Physics D Halliday,R Resnick, J Walker John Wiley And Sons,(1997)
4. For The Love Of Physics Walter Levin Free Press,(2011)
5. Concepts Of Modern Physics A Beiser, S Mahajan, Mc Graw Hill,(2010).

Course Code: CSB-3115 **Credit: 3**
Course Name: Basic Abstract & Linear Algebra

Course Objectives:

- Provide a foundational understanding of abstract algebraic structures such as groups, rings, and fields.
- Introduce students to linear algebra concepts including vector spaces, matrices, and linear transformations.
- Teach fundamental operations and properties of algebraic structures and matrices.
- Develop proficiency in solving linear systems of equations and performing matrix operations.
- Explore applications of abstract and linear algebra in various fields such as physics, engineering, and computer science.

Course Outcomes:

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

1. Recall the basic concepts of groups and their applications.
2. Explain the significance of the notions of Cosets and Lagrange's Theorem.
3. Compute the expression of permutation groups by using permutation Group.
4. Recall and apply definitions of vector spaces, subspaces, linear transformations, etc.
5. Identify and recognize common vector operations and their properties.

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. (2013). 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: CSB-3151 **Credit: 2**
Course Name: Software Engineering Lab

Course Objective:

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

Course Outcomes:

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

1. Understand the fundamental principles and concepts of software engineering, including requirements engineering, design, testing, and maintenance.
2. Apply software engineering methodologies such as Agile, Waterfall, and DevOps to develop and manage software projects efficiently.
3. Demonstrate proficiency in using various software engineering tools and technologies for version control, testing, and project management.
4. Develop critical thinking and problem-solving skills through hands-on experience with real-world software engineering scenarios and case studies.
5. Collaborate effectively in teams to design, implement, and deliver software solutions, demonstrating professionalism and adherence to best practices in software engineering.

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 case 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 case?

Lab 11. Write test cases 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. (2015). Software Engineering (10th ed.). Pearson.
2. Pressman, Roger S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.
3. Martin, Robert C. (2009). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
4. Beck, Kent. (2000). Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley.
5. Fowler, Martin. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

Semester – VI

Course Code: CSB–3211 **Credit: 4**
Course Name: IoT & Cloud Computing

Course Objective:

- Introduce students to the concepts and technologies of the Internet of Things (IoT) and cloud computing.
- Teach the architecture and components of IoT systems and cloud platforms.
- Explore protocols, standards, and communication technologies used in IoT and cloud environments.
- Provide hands-on experience in developing and deploying IoT applications on cloud platforms.
- Investigate security, privacy, and scalability challenges in IoT and cloud computing, along with strategies for addressing them.

Course Outcomes:

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

1. Students will gain a deep understanding of how IoT devices integrate with cloud computing platforms, including the architecture, communication protocols, and cloud-based services used in IoT applications.
2. Students will develop practical skills in designing, implementing, and managing IoT applications deployed on cloud platforms, including data collection, processing, and visualization.
3. Students will be able to identify and address security and privacy challenges in IoT systems, implementing best practices for securing IoT devices and data transmitted to and from cloud platforms.
4. Students will become proficient in utilizing cloud computing services such as storage, databases, serverless computing, and machine learning to enhance the functionality and scalability of IoT applications.
5. Students will be able to analyze emerging trends in IoT and cloud computing, such as edge computing, blockchain integration, and ethical considerations, and evaluate their potential impact on future IoT deployments and applications.

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, Case studies showcasing 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
2. "Edge Computing: A Primer" by AgustinusBorgyWaluyo
3. "Fog and Edge Computing: Principles and Paradigms" edited by Rajkumar Buyya, Satish Narayana Srirama, and Shadi Ibrahim

Course Code: CSB–3212 **Credit: 4**
Course Name: Machine Learning using Python

Course Objectives:

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

Course Outcomes:

1. Gain a solid understanding of the core principles and techniques of machine learning.
2. Develop practical skills in implementing and evaluating machine learning algorithms using Python libraries.
3. Apply machine learning algorithms to analyze and solve real-world problems in various domains.
4. Interpret and communicate the results of machine learning models effectively.
5. Prepare for further studies or professional roles in machine learning and data science.

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 AurélienGéron -
2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
3. "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido
4. "Deep Learning with Python" by François Chollet
5. "Pattern Recognition and Machine Learning" by Christopher M. Bishop

Course Code: CSB–3213 **Credit: 4**
Course Name: Discrete Mathematics

Course Objective:

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

Course Outcomes:

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

1. Understand and apply fundamental concepts of logic, including propositional and predicate logic, to analyze and construct logical arguments and expressions.
2. Utilize set theory concepts, such as set operations, relations, functions, and cardinality, to model and solve problems in computer science and information technology.
3. Apply combinatorial techniques, including permutations, combinations, and counting principles, to analyze and solve problems related to data structures, algorithms, and cryptography.
4. Analyze and solve graph theory problems, including graph representation, connectivity, shortest path algorithms, and graph traversal techniques, to model and solve real-world problems.
5. Demonstrate proficiency in algebraic structures, including groups, rings, and fields, and apply algebraic properties to analyze and solve problems in cryptography, coding theory, and computer security.

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, 5th Edition, 2006.
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, Sept. 2007.
4. J. P. Tremblay, R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill Pub, 1975.
5. Narsing Deo: Graph Theory and Applications

Course code: CSB-3214 **Credit: 3**

Course Name: Concepts of Physics

Course Objectives:

- Introduce students to the fundamental principles and laws of physics.
- Teach key concepts such as motion, forces, energy, and momentum.
- Develop an understanding of classical mechanics and its applications in describing the motion of objects.
- Explore basic principles of electricity and magnetism.
- Foster critical thinking skills through problem-solving exercises and hands-on experiments to illustrate physics concepts.

Course Outcomes:

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

1. To comprehend the scope of physics, the role of measurement and mathematics inphysics.
2. To infer the concepts of mechanics, electricity and magnetism, optics, andwaves.
3. To explain behavior of fluid and heat.

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 andelectric potential, Conductors and insulators, Electric circuits

Unit-5: Magneto-statistics: 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 inmotion,

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,(2009)

2. Introduction to Physics John D Cutnell. K. W. Johnson, D Young, S Stadler Wiley India Edition, (2015)
3. Fundamentals of Physics D Halliday, R Resnick, J Walker John Wiley And Sons(1997)
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: CSB–3251 **Credit: 4**
Course Name: Machine Learning Using Python Lab

Course Objective:

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

Course Outcomes:

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

1. Students will gain a solid understanding of fundamental machine learning concepts, including supervised and unsupervised learning, feature engineering, model evaluation, and deployment.
2. Students will develop practical skills in using Python libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn for data manipulation, visualization, and building machine learning models.
3. Students will be able to implement a variety of machine learning algorithms, including linear regression, logistic regression, decision trees, support vector machines, clustering algorithms, and neural networks.
4. Students will learn how to evaluate the performance of machine learning models using appropriate metrics and techniques and interpret the results to make informed decisions.
5. Students will apply machine learning techniques to solve real-world problems in various domains such as finance, healthcare, marketing, and natural language processing, gaining practical experience in developing machine learning solutions.

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. Bishop, Christopher M. "Pattern Recognition and Machine Learning." Springer, 2006.

Course Code: CSB–3291 **Credit: 3**
Course Name: Internship

Course Objectives:

- Provide students with practical exposure to the industry and real-world work environments.
- Allow students to apply theoretical knowledge gained during their BCA program to practical projects and tasks.
- Develop professional skills such as teamwork, communication, problem-solving, and time management.
- 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.
- Facilitate networking opportunities and professional connections that can potentially lead to job placements or future career opportunities after completing the internship.

Course Outcomes:

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

1. Demonstrate the ability to apply theoretical concepts and principles learned in their BCA coursework to real-world projects and tasks, effectively bridging the gap between academic learning and practical application.
2. Acquire hands-on experience in various aspects of computer applications, including software development, system administration, database management, cybersecurity, and IT support, enhancing their technical skills and proficiency in industry-standard tools and technologies.
3. Cultivate essential workplace skills such as teamwork, communication, problem-solving, adaptability, and time management through active participation in internship projects and interactions with colleagues and supervisors.
4. Establish meaningful connections with industry professionals, mentors, and peers, expanding their professional network and gaining insights into industry trends, emerging technologies, and career opportunities within the IT sector.
5. Reflect on their internship experiences, assess their strengths, weaknesses, and areas for improvement, and develop a personalized career development plan that aligns with their interests, goals, and aspirations in computer applications or related fields.

Semester – VII

Course Code: CSB–4111 **Credit: 4**
Course Name: Soft Computing

Course Objectives:

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

Course Outcomes:

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

1. Students will demonstrate a thorough understanding of Soft Computing principles, paradigms, and techniques, including fuzzy logic, neural networks, evolutionary algorithms, and swarm intelligence.
2. Students will be able to apply Soft Computing techniques to solve a wide range of real-world problems, including pattern recognition, optimization, decision making, and data analysis, by selecting appropriate algorithms and methodologies.
3. Students will be capable of critically analyzing and evaluating the performance and suitability of different Soft Computing techniques for specific tasks, considering factors such as computational complexity, robustness, and interpretability.
4. Students will gain practical experience in implementing Soft Computing algorithms and systems using programming languages and software tools commonly used in the field, enabling them to translate theoretical concepts into working solutions.
5. Students will recognize and explore the interdisciplinary applications of Soft Computing in diverse domains such as image processing, robotics, finance, bioinformatics, and quantum computing, fostering creativity and innovation in problem-solving approaches.

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

2. "Soft Computing and Intelligent Systems: Theory and Applications" by Madan M. Gupta and Narsingh Deo
3. "Introduction to Soft Computing" by Eva Volná, JánŠtepán, and V ladik Kreinovich
4. "Soft Computing: Techniques and Applications" edited by K. Vijayakumar, V. Manickavasagam, and M. Marikkannu
5. "Handbook of Soft Computing for Engineers and Scientists" edited by F. Castillo, P. Melin, and O. Montiel
6. "Soft Computing: New Trends and Applications" edited by J.M. Benítez, O. Cordón, F. Hoffmann, and R. Roy

Course Code: CSB–4112 **Credit: 4**
Course Name: R programming

Course Objective:

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

Course Outcomes:

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

1. **Proficiency in R Programming:** Students will demonstrate proficiency in R programming, including data manipulation, visualization, statistical analysis, and reporting, through practical assignments and projects.
2. **Data Manipulation Skills:** Students will be able to manipulate and transform data efficiently using R, including tasks such as cleaning, reshaping, merging, and summarizing datasets for analysis.
3. **Data Visualization Abilities:** Students will develop the ability to create insightful visualizations using R, including exploratory plots, statistical graphics, and interactive visualizations, to communicate patterns and insights in data effectively.
4. **Statistical Analysis Competence:** Students will acquire the skills to perform basic statistical analysis in R, including hypothesis testing, regression modeling, and descriptive statistics, to derive meaningful insights from data and make informed decisions.
5. **Reproducible Research Practices:** Students will learn and apply principles of reproducible research, including documenting and organizing code, generating dynamic reports with R Markdown, and creating reproducible workflows, to ensure transparency and replicability in data analysis and reporting.

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
2. "Advanced R" by Hadley Wickham
3. "The Art of R Programming: A Tour of Statistical Software Design" by Norman Matloff
4. "R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics" by Paul Teetor
5. "Introduction to Statistical Learning with Applications in R" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
6. "ggplot2: Elegant Graphics for Data Analysis" by Hadley Wickham

Course Code: CSB–4113 **Credit: 4**
Course Name: Network Security & Cryptography

Course Objective:

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

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

1. Students will demonstrate a thorough understanding of fundamental network security concepts, including threat analysis, risk assessment, access control mechanisms, and security policies, enabling them to identify and address security vulnerabilities in network infrastructures.
2. Students will acquire proficiency in cryptographic techniques, including symmetric and asymmetric encryption, hash functions, digital signatures, and cryptographic protocols, allowing them to secure data communication and ensure the confidentiality, integrity, and authenticity of information exchanged over networks.
3. Students will develop the ability to implement and configure security mechanisms, such as firewalls, intrusion detection systems (IDS), virtual private networks (VPNs), and secure authentication protocols, to protect network resources and mitigate security risks effectively.
4. Students will gain skills in security management practices, including risk assessment, security policy development, security awareness training, and incident response planning, enabling them to proactively manage security risks and respond effectively to security incidents and breaches.
5. Students will demonstrate awareness of legal and ethical considerations in network security practices, including compliance with relevant laws and regulations, protection of user privacy rights, and adherence to ethical standards of conduct, ensuring responsible and ethical behavior in security-related activities.

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

Course Code: CSB–4114 **Credit: 4**
Course Name: Parallel Computing

Course Objectives:

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

Outcomes:

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

1. Understand parallel computing concepts, including architectures, programming models, and performance metrics.
2. Develop proficiency in parallel programming paradigms, such as shared memory, distributed memory, and GPU computing.
3. Design, analyze, and optimize parallel algorithms for various computational problems.
4. Demonstrate effective problem-solving skills in parallel computing through practical assignments and projects.
5. Prepare for advanced studies and research in parallel computing, with a solid foundation in theory and practical experience.

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

2. "Parallel Programming: Concepts and Practice" by Barry Wilkinson and Michael Allen -
3. "Parallel Computing: Principles and Practice" by Rob H. Bisseling
4. "Parallel Programming in C with MPI and OpenMP" by Michael J. Quinn
5. "CUDA by Example: An Introduction to General-Purpose GPU Programming" by Jason Sanders and Edward Kandrot

Course Code: CSB-4115 **Credits:4**
Course Name: Linear Programming

Course Objectives:

- Introduce students to the theory and applications of linear programming, which involves optimizing a linear objective function subject to linear constraints.
- Teach the formulation of mathematical models as linear programming problems, including decision variables, objective functions, and constraints.
- Explore methods for solving linear programming problems, such as the simplex method, interior point methods, and graphical methods.
- Provide hands-on experience with linear programming software packages such as Excel Solver, GNU Linear Programming Kit (GLPK), and MATLAB Optimization Toolbox.
- 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:

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

1. Analyze and solve linear programming problem with graphicalmethod.
2. Discuss simplex algorithm to solve linear programmingproblems.
3. Determine the concept of duality in linear programming problems to get its solution using dual simplex method.
4. Apply Hungarian method to solve assignmentproblems.
5. Apply basic feasible solution techniques to solve transportation problem and get its optimal solution using modifiedmethod.

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 cases 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.
2. “Linear Programming”, G. Hadley, Oxford, IBH publishing Co.
3. “Operations Research”, P.K. Gupta, S. Chand & Company Ltd.

Semester – VIII

Course Code: CSB–4211 **Credit: 4**
Course Name: Natural Language Processing

Course Objectives

- Introduce students to the field of natural language processing, which involves the interaction between computers and human languages.
- Teach fundamental concepts such as tokenization, stemming, lemmatization, part-of-speech tagging, and syntactic parsing.
- Explore techniques for text classification, sentiment analysis, named entity recognition, and machine translation.
- Provide hands-on experience with NLP libraries and tools such as NLTK (Natural Language Toolkit), spaCy, and TensorFlow.
- 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

By the end of this course, students will be able to:

1. Students will effectively normalize and tokenize text, handle various linguistic nuances, and utilize regular expressions for text matching.
2. Students will be able to apply key NLP algorithms such as Bag-of-Words, TF-IDF, word embeddings, and text classification methods, and evaluate their performance using appropriate metrics.
3. Students will gain hands-on experience in building and fine-tuning advanced models, including RNNs, LSTMs, Transformers, and pre-trained language models like BERT and GPT.
4. Students will be capable of leveraging NLP techniques for practical applications, such as sentiment analysis, named entity recognition, machine translation, and dialogue systems, demonstrating the ability to address and solve complex language-related challenges.
5. Students will critically evaluate ethical considerations, such as bias and fairness in NLP models, and stay informed about recent advancements and future directions in the field, ensuring they are prepared to contribute responsibly to the evolving landscape of NLP.

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 (lowercasing, 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 cases 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
2. "Foundations of Statistical Natural Language Processing" by Christopher D. Manning and HinrichSchütze
3. "Natural Language Processing in Action" by Lane, H., Howard, H., and Hapke, H.
4. "Speech and Language Processing: Algorithms and Systems" by Mohamed Zakaria Kurdi and Khalid Shaalan
5. "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain, and Karan Kumar
6. "Applied Natural Language Processing: Identification, Investigation, and Resolution" by John P. McCrae and Sergio Roa

Course Code: CSB–4212 **Credit: 4**
Course Name: Web Designing

Course Objectives

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

Course Outcomes

By the end of this course, students will be able to:

1. Students will effectively use HTML for structuring web content, CSS for styling, and JavaScript for interactive functionalities, mastering the core technologies of web development.
2. Students will design websites that adapt seamlessly to various devices and screen sizes, ensuring accessibility for all users by following best practices and standards.
3. Students will gain hands-on experience with design tools (e.g., Adobe XD, Figma), front-end frameworks (e.g., Bootstrap, React), and CSS preprocessors (e.g., Sass), enhancing their design workflow and productivity.
4. Students will apply techniques to improve website loading speed, performance, and search engine ranking, using tools and methodologies to test and enhance web performance.
5. Students will understand and be able to implement cutting-edge web technologies and trends, such as Progressive Web Apps (PWAs), Single Page Applications (SPAs), and WebAssembly, ensuring they remain competitive in the dynamic field of web design.

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

Course Code: CSB–4213 **Credit: 4**

Course Name: Data Warehousing and Data Mining

Course Objectives

- 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.
- Teach data mining techniques for extracting valuable insights and patterns from data stored in data warehouses.
- Explore data preprocessing methods such as data cleaning, transformation, and integration to prepare data for analysis.
- Provide hands-on experience with data mining algorithms and tools such as association rule mining, classification, clustering, and anomaly detection.
- Foster critical thinking skills by applying data warehousing and data mining techniques to real-world datasets and business problems.

Course Outcomes

By the end of this course, students will be able to:

1. Develop and implement efficient data warehousing solutions, including schema design, ETL processes, and data integration.
2. Apply various data mining algorithms for classification, clustering, association, and other data mining tasks to discover patterns and insights from large datasets.
3. Understand the synergies between data warehousing and data mining, and utilize data warehouses as a source for data mining processes.
4. Identify and address ethical, privacy, and security issues related to data warehousing and mining, ensuring compliance with relevant laws and regulations.
5. Use data warehousing and data mining techniques to solve real-world problems in various domains such as marketing, finance, healthcare, and telecommunications.

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

Course Code: CSB–4214 **Credit: 4**
Course Name: Digital Forensics

Course Objective:

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

Course Outcomes:

By the end of this course, students will be able to:

1. Students will demonstrate proficiency in using a variety of digital forensics tools and techniques for acquiring, analyzing, and interpreting digital evidence from diverse sources such as storage devices, networks, and memory dumps.
2. Students will develop an understanding of the legal and ethical considerations that govern digital forensics investigations, including the admissibility of digital evidence in court, privacy rights, and chain of custody requirements.
3. Students will acquire skills in incident response and handling, including identifying security incidents, containing threats, eradicating malware, and restoring systems to a secure state to minimize damage and data loss.
4. Students will enhance their critical thinking and problem-solving abilities by analyzing complex digital evidence, identifying patterns and anomalies, and drawing conclusions to support investigative findings and decision-making.
5. Students will develop effective communication and reporting skills to convey forensic findings, methodologies, and conclusions clearly and concisely to technical and non-technical stakeholders, including law enforcement agencies, legal professionals, and organizational management.

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 EnCase 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, Case studies and landmark legal decisions in digital forensics

References:

1. "Computer Forensics and Cyber Crime: An Introduction" by Marjie T. Britz
2. "Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet" by Eoghan Casey
3. "File System Forensic Analysis" by Brian Carrier
4. "Guide to Computer Forensics and Investigations" by Bill Nelson, Amelia Phillips, and Christopher Steuart
5. "Handbook of Digital Forensics and Investigation" edited by Eoghan Casey
6. "Investigating Digital Crime" by Ken Anderson, Rob Lee, Jamie Levy, and Kris Kendall

Course Code: CSB–4215 **Credit: 4**

Course Name: Bioinformatics Using AI

Course Objective:

- Introduce students to the interdisciplinary field of bioinformatics, which involves the application of computational techniques to analyze biological data.
- Teach fundamental concepts in biology, genetics, and genomics relevant to bioinformatics analysis.
- Explore artificial intelligence (AI) techniques such as machine learning, deep learning, and natural language processing for analyzing biological data.
- Provide hands-on experience with AI tools and frameworks used in bioinformatics, such as TensorFlow, Keras, and scikit-learn.
- 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:

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

1. Gain a thorough understanding of the principles and applications of Bioinformatics and AI in computational research, enabling effective utilization of AI techniques for biological data analysis.
2. Develop proficiency in data preprocessing techniques, feature engineering, and dimensionality reduction methods tailored for computational data in bioinformatics.
3. Master various supervised and unsupervised machine learning algorithms, including ensemble methods and deep learning architectures, for classification, regression, clustering, and pattern discovery in biological datasets.
4. Acquire skills in structural bioinformatics, research data analysis, and interpretation using AI techniques, empowering students to contribute to cutting-edge research in the field.
5. Demonstrate the ability to integrate diverse biological datasets, formulate system models, and apply AI approaches to address complex biological questions, fostering interdisciplinary collaboration and innovation in bioinformatics research.

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. Mount, D. W. (2004). Bioinformatics: Sequence and genome analysis. Cold Spring Harbor Laboratory Press.
2. Russell, S., &Norvig, P. (2021). Artificial intelligence: A modern approach. Pearson.
3. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
4. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
5. James, G., Witten, D., Hastie, T., &Tibshirani, R. (2013). Introduction to statistical learning: With applications in R. Springer.

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 (Open and Distance Learning Programmes) Regulations, 2020.

Syllabi and Course Materials

The Syllabi, PPR and Self-Learning Materials (SLMs) are developed mostly by experienced faculty members of MATA TRIPURA SUNDARI OPEN 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. The course material prepared by the University faculty is at par with the regulations2020.

List of Faculty associated with BCA programme is as follows:

S.No.	Name of Faculty	Designation	Nature of Appointment	Qualification	Subject
1.	Dr. Manoj Varshney	Professor	Full-Time	Ph.D.	Computer Science
2.	Dr. Sir Mishra Roy	Associate Professor	Full-Time	Ph.D.	Computer Science
3.	Dr. Ankur Kumar	Assistant Professor	Full-Time	Ph.D.	Computer Science
4.	Dr. Manish Saxena	Assistant Professor	Full-Time	Ph.D.	Computer Science
5.	Dr. Duvvuri B. K. Kamesh	Assistant Professor	Full-Time	Ph.D.	Computer Science

Delivery Mechanism

The MTSOU follows a modern ICT (Information & Communication Technology) enabled approach for instruction. The methodology of instruction in this university is different from that of the conventional/regular programs. Our system is more learner-oriented and the learner is an active participant in the teaching-learning process. MTSOU 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 course will be held on the campus on Saturday and on Sunday of 2 hour 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.

C. Medium of Instruction

Medium of Course Instruction: English

Medium of Examination: English

D. Student Support Systems

Universities Study Centres 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 of distance mode.

The University is not promoting any study centres outside the State of Tripura. All student support services will be provided to the student through the University Headquarters and the study centres.

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 MTSOU 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	13500	2000	15500
			2	12000	2000	14000
			3	12000	2000	14000
			4	12000	2000	14000

Total 57500

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, practicals, etc.)	Sep	Nov	Mar	May

Credit System

MTSOU 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 8Yrs	160	BCA	Bachelor's Degree

Assignments

Distance Education learners have to depend much on self study. 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 Study Centre 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 (www.mtsou.edu.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.

G. Laboratory Support and Library Resources

The library of MATA TRIPURA SUNDARI OPEN 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 MATA TRIPURA SUNDARI OPEN UNIVERSITY has a dedicated Library for programs and acquiring printed books and e-books for this purpose. The required International and National subject journals are also provided.

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 etc.

H. 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	:	60%
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.

I. 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 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 Open and Distance 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 programmes and matches with the quality of relevant programmes 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 programmesin 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.

After enrolling in BCA Programme at MATA TRIPURA SUNDARI OPEN UNIVERSITY, student will exhibit understanding in areas such as critical thinking, effective communication and develop problem solving, scientific temperament with right set of ethics and attitude towards environment and sustainability. After completion of BCA Programme, student will participate in multiple functional areas of science and technology.