

126CAT01 MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

COURSE OBJECTIVES:

- To introduce set theory and functions.
- To expose the concept of logic and hence to construct valid mathematical arguments.
- To master combinatorics which deals with the counting principles.
- To introduce formal languages and finite state automata.
- To learn network scheduling

UNIT – I FUNDAMENTAL STRUCTURES 9+3

Set theory- Relationships between sets – Operations on sets – Set identities – Principle of inclusion and exclusion – Minsets- Relations- Binary relations – Partial orderings – Equivalence relations-Functions- Properties of functions – Composition of functions – Inverse functions.

Case study

Mathematical Modeling of Social Networks using Sets and Relations

UNIT – II LOGIC 9+3

Propositional logic – Logical connectives – Truth tables – Normal forms (conjunctive and disjunctive) – Predicate logic – Universal and existential quantifiers – Proof techniques(direct and indirect).

Case study

Logical Reasoning in Automated Theorem Proving

UNIT – III COMBINATORICS 9+3

Mathematical Induction – Strong induction and well-ordering- Basics of counting – Counting arguments – Pigeonhole principle – Permutations and Combinations – Recurrence relations- Solving linear recurrence relations – Generating functions.

Case study

Counting Techniques in Combinatorial Optimization

UNIT – IV FORMAL LANGUAGES& FINITE AUTOMATA 9+3

Formal Languages – Classes of Grammars – Phrase structure – Context sensitive – Context Free – Regular Grammars – Finite State Automata –Non- Deterministic Finite State Automata – Deterministic Finite State Automata.

Case study

Mathematical Foundations of Pattern Recognition using Finite Automata

UNIT – V NETWORK SCHEDULING BY PERT AND CPM**9+3**

Network construction- Critical path method(CPM)-time charts and resource leveling- Project evaluation and review technique(PERT)- resource analysis in network scheduling-Minimal spanning tree-Shortest route.

Case study

Project Scheduling and Optimization Using Mathematical Network Models

Total No. of Periods : 45 + 15

Note : Use of approved statistical table is permitted in the examination.

COURSE OUTCOMES:

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

- CO1 :** Use the set theory concepts to understand the structures of functions. (K2)
- CO2 :** Expertise the knowledge of logics helps to verify the correctness of computer programs and to draw conclusions from scientific experiments. (K3)
- CO3 :** Demonstrate the applications of basic concepts of an algorithm and counting principles in combinatorial mathematics. (K3)
- CO4 :** Construct finite automata machines using automata theory and formal languages. (K4)
- CO5 :** Apply optimum network scheduling using PERT and CPM. (K3)

REFERENCES:

1. Judith L.Gersting, “Mathematical Structures for Computer Science”, W.H.Freeman and Company, NY, 2012.
2. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, seventhth edition, TMH, 2013.
3. M.K. Venkataraman, N. Sridharan and N.Chandrasekaran, “ Discrete Maths.”, The National Publishing Company, 2012.
4. Hopcraft.J.E., Motwani.R., Ullman.J.D., Introduction to Automata Theory, Languages and Computations”, 3rd Edition, Pearson Education, 2008.
5. John.C.Martin, “Introduction to Languages and Theory of Computation”, 4th edition, Tata McGraw Hill, 2011.
6. Hamdy A Taha, “Operations Research – An Introduction”, Pearson, 10th Edition, 2016.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAT01 MATHEMATIC AL FOUNDATIONS OF COMPUTER SCIENCE	CO1	3	3	2	2	1	1	1	2	3	2	1
	CO2	3	3	2	3	1	1	1	2	3	2	1
	CO3	3	3	3	3	2	1	1	2	3	2	2
	CO4	3	2	3	3	2	1	1	2	3	2	2
	CO5	3	2	2	3	2	1	1	2	3	3	2
Average		3.00	2.60	2.40	2.80	1.60	1.00	1.00	2.00	3.00	2.20	1.60

126CAT02 ADVANCED DATA STRUCTURES AND ALGORITHMS

BASIC PREREQUISITES:

- Knowledge in C Structure
- Understanding of Pointers
- Concept in Array and Linked List.

COURSE OBJECTIVES:

- To explore the principles of algorithm efficiency and complexity in computing.
- Understand and Hierarchical data structures and their applications.
- To Master fundamental graph algorithms and their applications
- Introduce dynamic programming and greedy strategies to solve optimization problems.
- Understand NP-completeness and its implications for computational complexity.

UNIT – I ROLE OF ALGORITHMS IN COMPUTING AND COMPLEX ANALYSIS 9

Algorithm – Algorithms as a Technology – Time and Space Complexity of Algorithms – Asymptotic Analysis – Average and Worst Case Analysis – Asymptotic Notation – Importance of Efficient Algorithms – Program Performance Measurement – Recurrences Substitution Method – The Recursion – Tree Method.

Case Study

A team developing an online food delivery application observed slow restaurant search and order allocation during peak hours. They analyzed the time and space complexity of linear search, binary search, and recursive algorithms to improve system performance and scalability.

UNIT – II HIERARCHIAL DATA STRUCTURES 9

Binary Search Trees: Basics – Querying a Binary Search Tree – Insertion and Deletion – Huffman Coding – Red Black Tree: Properties – Insertion – Deletion- Rotation – B-tree: Definition- Basic Operation – Heap – Heap Implementation - Disjoint Sets – Fibonacci Heaps: Structure – Merge able – Heap Operations – Decreasing a Key and Deleting a node - Bounding the maximum degree.

Case Study

A digital library system used Binary Search Trees, Red-Black Trees, B-Trees, and Heaps to improve fast data storage, search, and retrieval performance.

UNIT – III GRAPHS 9

Elementary Graph Algorithms: Representation of Graphs – Breadth First and Depth First Search Topological Sort – Minimum Spanning Tree: Prim's – Kruskal- Single Source Shortest Path: Bellman – Ford Algorithm – Single Source Shortest path in Directed Acyclic Graph: Dijkstra's Algorithm- Dynamic Programming: The Floyd – Warshall Algorithms.

Case Study

A navigation and transportation system used graph algorithms such as BFS, DFS, Dijkstra's, Prim's, and Kruskal's algorithms to find shortest paths, optimal routes, and minimum network costs efficiently.

UNIT – IV ALGORITHM DESIGN TECHNIQUES

9

Dynamic Programming: Matrix – Chain Multiplication – Elements of Dynamic Programming – Longest Common Subsequence – Greedy Strategy – An Activity – Selection Problem.

Case Study

A scheduling and data processing system used Dynamic Programming and Greedy algorithms to optimize matrix operations, identify common data patterns, and efficiently allocate activities and resources.

UNIT – V NP COMPLETE AND NP HARD

9

NP – Completeness: Polynomial Time – Time Verification – NP Completeness and Reducibility
NP Completeness Proofs – NP Complete Problem.

Case Study

A logistics company analyzed NP-Complete problems to optimize delivery scheduling and route planning, where finding the exact optimal solution required high computational time for large datasets.

Total No. of Periods : 45

COURSE OUTCOMES:

At the end of course- the Student will be able to:

- CO1 :** Analyze e algorithm performance using various methods
- CO2 :** Manipulate advanced data structures like trees and heaps
- CO3 :** Implement and Analyze key graph algorithms to solve real-world problems
- CO4 :** Apply dynamic programming and greedy methods to solve real-world problems efficiently.
- CO5 :** Identify and prove NP-complete problems and analyze their complexity

REFERENCE BOOKS :

1. Thomas H. Cormen- Charles E. Leiserson- Ronald L. Rivest- Clifford Stein-Introduction to Algorithms- MIT Press- 2023 (4th edition).
2. Robert Sedgewick- Kevin Wayne- Algorithms- Addison-Wesley- 2022 (4th edition).
3. Steven S. Skiena- The Algorithm Design Manual- Springer- 2020 (3rd edition).
4. Mark Allen Weiss- Data Structures and Algorithm Analysis in C++- Pearson- 2019(5th edition).
5. Robert Lafore- Advanced Data Structures and Algorithms in Java- Sams Publishing-2021 (3rd edition).
6. Narasimha Karumanchi- Data Structures and Algorithms Made Easy- CareerMonkPublications- 2019 (5th edition).
7. Jeffrey D. Ullman- John E. Hopcroft- Introduction to Automata Theory- Languages-and Computation- MIT Press- 2020 (3rd edition).
8. S. Sridhar- “Design and Analysis of Algorithm”- Oxford University Press- 1 st Edition- 2014

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAT02 ADVANCED DATASTRUCTURES AND ALGORITHMS	CO1	3	3	3	3	1	1	1	2	3	2	1
	CO2	3	3	3	3	2	1	1	2	3	2	1
	CO3	3	3	3	3	2	1	1	2	3	2	1
	CO4	3	3	3	3	2	1	1	2	3	2	1
	CO5	3	3	3	2	1	1	1	2	2	2	1
Average		3.00	3.00	3.00	2.80	1.60	1.00	1.00	2.00	2.80	2.00	1.00

126CAT03 RESEARCH METHODOLOGY AND IPR

BASIC PREREQUISITES:

- Basic knowledge of Data Collection Methods
- Basic understanding of Hypothesis
- Basic knowledge of Report Writing

COURSE OBJECTIVES:

- To understand the principles of experimental design and sampling methods.
- To familiarize students with various measurement scales, data collection techniques, and the development of measurement tools.
- To provide a foundation for formulating and testing hypotheses using parametric and non-parametric statistical methods.
- To understand the principles of Intellectual Property Rights, their evolution, and their implications for innovation and research.
- To analyze the objectives, types, and procedures involved in the patenting process and develop strategies for acquiring and managing patents effectively.

UNIT - I RESEARCH DESIGN

6

Meaning of Research Design – Need for Research Design – Features of Good Design – Different Research Design – Basic Principles of Experimental Design – Census and Sample Survey – Implications of a Sample Design – Steps in Sampling Design – Criteria for Selecting a sample procedure – Characteristics of Good Sample Design – Different types of Sampling Designs.

Case Study

Sampling and research design for evaluating a software application's usability.

UNIT – II MEASUREMENT AND METHODS FOR DATA COLLECTION

6

Measurement in Research – Scales – Error Measurement – Types of Measurement – Techniques for Developing Measurement tools – Scaling – Meaning of Scaling – Bases for Classification of Scales – Scaling Techniques – Collection of Primary Data – Observation – Interview Method – Collection of Data Through Questionnaires – Data Through Schedules – Difference between Questionnaires and Schedules.

Case Study

Questionnaire and interview methods for collecting user requirements.

UNIT – III HYPOTHESIS TESTING

6

Hypothesis – Basic Concepts – Procedure – Flow Diagram – Tests of Hypotheses – Parametric Tests – Hypothesis Testing: Means – Difference between Means – Comparing Two Related Samples – Proportions – Difference between Proportions – Comparing a Variance – Correlation Coefficients – Non-Parametric Tests – Chi-Square Test – Mann-Whitney U Test – Wilcoxon Signed-Rank Test – Limitations of Hypothesis Testing.

Case Study

Statistical testing of machine-learning or software-performance results

UNIT – IV INTELLECTUAL PROPERTY RIGHTS

6

Intellectual Property – Concept of IPR – Evolution and Development of IPR – IPR Development Process – Copyright – Trade Secrets – Utility Models – IPR and Biodiversity – Right of Property – Common Rules of IPR Practices – Types and Features of IPR Agreements – Trademarks – Role of WIPO and WTO in IPR – TRIPS Agreement.

Case Study

Copyright, patents, trademarks and trade secrets in software/IT.

UNIT - V PATENTS

6

Patents – objectives and benefits of patent- Concept- features of patent- Inventive step- Specification- Types of patent application- process E-filing - Examination of patent- Grant of patent- Revocation- Assignment – Licensing – Licensing of Related Patents - patent agents- Registration of patent agents

Case Study

Patent filing for a software/hardware-based technological innovation.

Total No. of Periods : 30

COURSE OUTCOMES:

At the end of course- the Student will be able to:

- CO1 :** Design and evaluate research studies using appropriate research designs and sampling techniques.
- CO2 :** Develop and implement appropriate measurement tools and data collection methods for research.
- CO3 :** Apply and interpret appropriate parametric and non-parametric statistical methods for testing research hypotheses.
- CO4 :** Apply key concepts of Intellectual Property Rights to protect and manage innovations and creative works.
- CO5 :** Explain the patent application process and analyze patent filing, examination, licensing, and management procedures.

REFERENCE:

1. Cooper Donald R- Schindler Pamela S and Sharma JK- “Business Research Methods”- Tata McGraw Hill Education- 11e (2012).
2. Catherine J. Holland- “Intellectual property: Patents- Trademarks- Copyrights- TradeSecrets”- Entrepreneur Press- 2007.
3. David Hunt- Long Nguyen- Matthew Rodgers- “Patent searching: tools & techniques”- Wiley- 2007.
4. The Institute of Company Secretaries of India- Statutory body under an Act of parliament- “Professional Programme Intellectual Property Rights- Law and practice”- September 2013.
5. C. B. Gupta, *Research Methodology: Concepts, Methods and Techniques*, 3rd Edition, New Age International Publishers, 2023.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAT03 RESEARCH METHODOLOG Y AND IPR	CO1	3	3	2	2	2	2	2	3	1	3	2
	CO2	3	3	2	2	2	1	2	3	1	3	1
	CO3	3	3	2	2	1	1	1	3	1	2	1
	CO4	2	2	2	1	2	2	3	2	1	2	3
	CO5	2	2	3	1	2	3	3	2	1	2	3
Average		2.60	2.60	2.20	1.60	1.80	1.80	2.20	2.60	1.00	2.40	2.00

126CAI01 DATASCIENCE WITH PYTHON PROGRAMMING

BASIC PREREQUISITES:

- Basic knowledge of Python programming- data handling- and elementary statistics.

COURSE OBJECTIVES:

- To understand the fundamentals of Python programming for data science applications.
- To learn data handling and pre-processing techniques using Python libraries.
- To apply data visualization and exploratory analysis methods for interpreting datasets.
- To understand basic statistical and machine learning concepts used in data science.
- To develop practical skills for solving real-world data analysis problems using Python.

UNIT – I INTRODUCTION TO PYTHON PROGRAMMING 9+6

Python Environment and Jupyter Notebook- Variables and Data Types- Operators and Expressions- Conditional Statements- Looping Statements- Functions and Modules- Strings and String Manipulation- Lists- Tuples- Dictionaries and Sets.

Lab Exercise

- Programs using conditional and looping statements.
- Programs using functions strings lists tuples dictionaries and sets.

Case Study

Automated Research Publication Analytics System – Develop Python scripts to process publication records- identify author-wise publication counts- citation metrics- and generate summary reports for academic departments.

UNIT – II NUMPY FUNDAMENTALS 9+6

Introduction to NumPy- ndarray Object- Array Creation and Indexing- Array Operations and Vectorization- Mathematical Functions- Statistical Operations- Random Number Generation- File Input and Output with NumPy.

Lab Exercise

- Array creation indexing slicing and mathematical operations using NumPy.
- Statistical analysis and matrix operations using NumPy arrays.

Case Study

Air Quality Data Analysis using NumPy – Analyze environmental sensor data to compute statistical measures- detect anomalies- and compare pollution levels across different regions.

UNIT – III DATA HANDLING USING PANDAS 9+6

Introduction to Pandas- Series and Data Frames- Data Import and Export- Data Cleaning- Handling Missing Values- Data Transformation- Sorting and Filtering- Group By Operations and Aggregation.

Lab Exercise

- Creation and manipulation of Series and DataFrames.
- Data cleaning filtering grouping and aggregation using Pandas..

Case Study

Healthcare Data Management and Analysis – Process patient records to identify disease trends- demographic patterns- and treatment outcomes using Pandas.

UNIT–IV DATA VISUALIZATION AND EXPLORATORY ANALYSIS 9+6

Introduction to Data Visualization- Matplotlib Basics- Line Bar Pie Histogram and Scatter Plots- Data Analysis using Pandas- Descriptive Statistics- Correlation and Covariance- Exploratory Data Analysis- Time Series Basics.

Lab Exercise

- Creation of line bar histogram and scatter plots using Matplotlib.
- Exploratory data analysis using descriptive statistics and correlation measures

Case Study

E-Commerce Sales Performance Dashboard – Analyze transaction data to visualize customer behavior- product performance- seasonal trends- and revenue patterns.

UNIT – V DATA SCIENCE APPLICATIONS

9+6

Data Pre-processing Techniques- Data Aggregation and Reshaping- Introduction to Machine Learning Concepts- Regression Basics- Classification Basics- Data Science Workflow- Case Studies using Python- Applications of Data Science.

Lab Exercise

- Data pre-processing and reshaping using Python libraries.
- Basic regression and classification using Scikit-learn.

Case Study

Predictive Analysis of Customer Churn – Develop a data-driven solution to identify customers likely to discontinue a service and recommend retention strategies based on analytical insights

Total No. of Periods : 45+30

COURSE OUTCOMES:

At the end of course- the Student will be able to:

- CO1:** Apply Python programming concepts for data science applications.
- CO2:** Perform data pre-processing and data handling using NumPy and Pandas libraries.
- CO3:** Create meaningful visualizations and perform exploratory data analysis.
- CO4:** Apply basic statistical and machine learning concepts for data analysis tasks.
- CO5:** Develop data science solutions using Python tools and libraries.

REFERENCES

1. Abraham Silberschatz- Henry F. Korth- S. Sudarshan - “Database System Concepts”- McGraw-Hill Education- 2019 .
2. Ramez Elamassri and Shankant B-Navathe- “Fundamentals of Database Systems”- Sixth Edition- Pearson Education Delhi- 2010.
3. Brad Dayley- “Teach Yourself NoSQL with MongoDB in 24 Hours”- Sams Publishing- First Edition- 2014.
4. Raghu Ramakrishnan- Johannes Gehrke -’ Database management systems”- McGraw-Hill- 6rd Edition- 2014.
5. Frank. P. Coyle- “XML Web Services and the Data Revolution”- Pearson Education- 2012.
6. Lee Chao- “Database Development and Management”- Auerbach Publications- 2016

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAI01 DATA SCIENCE WITH PYTHON PROGRAMMING	CO1	3	2	2	2	1	1	1	2	3	2	1
	CO2	3	3	2	2	1	1	1	2	3	2	1
	CO3	3	2	3	2	2	1	1	2	3	2	2
	CO4	3	2	3	3	2	1	1	2	3	3	2
	CO5	3	2	3	3	2	1	1	2	3	3	2
Average		3.00	2.20	2.60	2.40	1.60	1.00	1.00	2.00	3.00	2.40	1.60

126CAI02 DATABASE MANAGEMENT SYSTEMS

BASIC PREREQUISITES:

- Basic knowledge of Database Concepts.

COURSE OBJECTIVES:

- To understand the concepts, architecture, design, and implementation of database management systems.
- To understand database design, SQL, and database programming techniques.
- To study the principles and architecture of distributed and parallel databases.
- To understand and manage database transactions, concurrency control, recovery, and security.
- To design and implement efficient file structures, hashing, indexing, and special-purpose databases including NoSQL databases.

UNIT – I INTRODUCTION

9+6

File Systems versus Database Systems – Data Models – DBMS Architecture – Data Independence – Data Modeling using Entity-Relationship Model – Enhanced E-R Modeling.

Lab Exercise

- DDL, DML and DCL commands
- Operators and built-in functions

Case Study

Design an ER Diagram for a Student Admission Management System including Students, Courses, Departments, and Admissions.

UNIT – II RELATIONAL MODEL AND QUERY EVALUATION

9+6

Relational Model Concepts – Relational Algebra – SQL – Basic Queries – Complex SQL Queries Joins -Views – Constraints – Partitions - Dynamic SQL, Other SQL Functions, overview of commercial RDBMSs – Database Design – Functional Dependencies – Normal Forms –1NF – 2NF-3NF-BCNF – 4NF-5NF - Algorithms for Executing Query Operations– Query Processing– Overview– Measures of Query Cost.

Lab Exercise

- Joins
- Views
- Constraints
- Table Partitioning
- Dynamic SQL
- Sequences
- Indexes
- Subqueries and Nested Queries

Case Study

A library maintains information about books, members, authors, and issue transactions. Design the relational schema and identify primary keys, foreign keys, and relationships.

UNIT – III TRANSACTION PROCESSING

9+6

Transaction Processing – Properties of Transactions – Serializability – Transaction support in SQL
Time Stamp ordering – Recovery concepts – Shadow paging, Buffer Management – Log Based
Recovery – Database Security Issues – Access control.

Lab Exercise

- GROUP BY and HAVING
- ORDER BY and Aliases
- Transaction Control using COMMIT, ROLLBACK and SAVEPOINT
- PL/SQL Decision-Making
- PL/SQL Loops
- Exception Handling

Case Study

Two customers simultaneously transfer money from the same account. Explain how transaction management prevents inconsistencies.

UNIT - IV FILES, INDEXING & SPECIAL PURPOSE DATABASES 9+6

File Structures – Disks – RAID – File Operations – Hashing Techniques – Indexing – Single-Level and Multi-Level Indexes – B+ Trees – Static Hashing – Indexes on Multiple Keys – Distributed Databases – Parallel Databases – Object-Relational Databases.

Lab Exercise

- PL/SQL Procedures with Parameters – Functions – Cursors – Triggers

Case Study

Design a database for hospital management system

UNIT - V NOSQL DATABASES

9+6

NoSQL – Need for NoSQL Databases – Characteristics and Types of NoSQL Databases – JSON and JSON Data Model – CAP Theorem – BASE Properties – Document-Based Databases – Key-Value Databases – Column-Family Databases – Graph Databases – MongoDB – MongoDB Architecture and Data Model – Databases, Collections and Documents – MongoDB Operations: Insert, Update, Delete and Query – Filtering and Projection – Sorting – Aggregation – Indexing – Replication – Sharding – Deployment – Advanced MongoDB Features – Cassandra – Cassandra Architecture – Data Model – Keyspace – Tables – CQL Data Types – CRUD Operations – Partitioning and Replication – Consistency Levels – Introduction to MongoDB Application Development using PHP/Java.

Lab Exercise

- Create, insert, update, delete, and query JSON documents using MongoDB.
- Implement CRUD operations using MongoDB.
- Implement MongoDB queries using filtering, projection, sorting, and aggregation.
- Create and use indexes in MongoDB.
- Perform basic CRUD operations using Cassandra/CQL.

Case Study

Design and implement a NoSQL-based e-commerce product catalogue using MongoDB. Perform CRUD operations, filtering, aggregation, and indexing, and explain the role of replication and sharding in supporting scalability.

Total No. of Periods : 45+30

COURSE OUTCOMES:

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

- CO1 :** Explain the fundamental concepts, architecture, and components of database management systems. (K2)
- CO2 :** Analyze relational models, design normalized databases, develop SQL queries, and evaluate query processing techniques. (K4)
- CO3 :** Apply transaction processing, concurrency control, recovery techniques, and database security mechanisms. (K3)
- CO4 :** Design and implement efficient file structures, hashing, indexing, and storage management techniques. (K4)
- CO5 :** Develop and deploy database applications using NoSQL database systems such as MongoDB and Cassandra. (K3)

REFERENCES

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan , “Database System Concepts”, McGraw-Hill Education, 2019 .
2. Ramez Elamassri and Shankant B-Navathe, “Fundamentals of Database Systems”, Sixth Edition, Pearson Education Delhi, 2010.
3. Brad Dayley, “Teach Yourself NoSQL with MongoDB in 24 Hours”, Sams Publishing, First Edition, 2014.
4. Raghu Ramakrishnan, Johannes Gehrke , ‘ Database management systems”, McGraw, Hill, 6th Edition, 2014.
5. Frank. P. Coyle, “XML Web Services and the Data Revolution”, Pearson Education, 2012.
6. Lee Chao, “Database Development and Management”, Auerbach Publications, 2016.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAI02 DATABASE MANAGEMENT SYSTEMS	CO1	3	2	2	2	1	1	1	2	3	2	1
	CO2	3	3	3	3	2	1	1	2	3	2	2
	CO3	3	3	3	3	2	2	2	2	3	2	2
	CO4	3	2	3	3	2	2	1	2	3	2	2
	CO5	3	2	3	3	2	2	1	2	3	3	2
Average		3.00	2.40	2.80	2.80	1.80	1.60	1.20	2.00	3.00	2.20	1.80

126CAP01 ADVANCED DATA STRUCTURES AND ALGORITHMS LAB

BASIC PREREQUISITES:

- Knowledge of Structures in C
- Understanding of Pointers in C
- Knowledge of Arrays and Linked Lists

COURSE OBJECTIVES :

- Learn to implement stacks, queues, and linked lists.
- Develop skills in managing binary search trees, red-black trees, and Fibonacci heaps
- Use algorithms for shortest paths and minimum spanning trees.
- Evaluate the time and space complexity of various algorithms.
- Implement algorithms like Floyd-Warshall and Warshall's for solving graph problems.

LIST OF EXPERIMENTS

1. Implement forward substitution and Backward Substitution.
2. Measure the execution time of any algorithms.
3. Create a Binary Search Tree with traversal.
4. Implement the operation of Red Black Tree
5. Create and Process Fibonacci Heaps
6. Implement Prims and Kruskal Algorithm
7. Find the Shortest Path Algorithm using Dijkstra Algorithms.
8. Generate Floyds Algorithm and Warshall Algorithm
9. Computation of polynomial time.
10. Implement the Subset Sum Problem and analyze its computational complexity with respect to the NP class

Total No. of Periods : 60

COURSE OUTCOMES:

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

- CO1 :** Proficient in implementing and using basic data structures. (K3)
- CO2 :** Skilled in handling complex tree and heap structures. (K3)
- CO3 :** Apply graph algorithms to solve practical problems. (K4)
- CO4 :** Analyze and optimize algorithm performance. (K4)
- CO5 :** Implementing algorithms for shortest paths and transitive closures. (K4)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAP01 ADVANCED DATASTORES AND ALGORITHMS LAB	CO1	3	3	3	3	2	1	1	3	3	2	2
	CO2	3	3	3	3	2	1	1	3	3	2	2
	CO3	3	3	3	3	2	1	1	3	3	2	2
	CO4	3	3	3	3	2	1	1	3	3	2	2
	CO5	3	3	3	3	2	1	1	3	3	2	2
Average		3.0	3.0	3.0	3.0	2.0	1.0	1.0	3.0	3.0	2.0	2.0

126CAP02 MINI PROJECT

COURSE OBJECTIVES :

- To apply the concepts learned in Mathematical Foundations of Computer Science, Advanced Data Structures and Algorithms, Data Science with Python Programming, and Database Management Systems to solve real-world problems.
- To develop analytical and problem-solving skills through the identification and formulation of practical computing problems.
- To design and implement software solutions using appropriate algorithms, data structures, programming techniques, and database technologies.
- To enhance project planning, teamwork, technical documentation, and presentation skills.
- To gain hands-on experience in developing, testing, and validating computing solutions in a professional environment.

Development of a software solution for a real-world problem by applying the concepts of Mathematical Foundations, Advanced Data Structures and Algorithms, Data Science with Python Programming, and Database Management Systems, including analysis, design, implementation, testing, documentation, and presentation.

COURSE OUTCOMES :

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

- CO1 :** Analyze real-world problems and identify appropriate computing solutions. (K4)
- CO2 :** Apply mathematical, algorithmic, programming, and database concepts in the development of software applications. (K3)
- CO3 :** Design and implement a software system using suitable tools, techniques, and technologies. (K6)
- CO4 :** Evaluate the performance and correctness of the developed solution through testing and validation. (K5)
- CO5 :** Prepare technical documentation and effectively present project outcomes while demonstrating teamwork and professional skills. (K6)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAP02 MINI PROJECT - I	CO1	3	3	3	3	2	1	1	2	3	2	2
	CO2	3	3	3	3	3	1	1	2	3	3	2
	CO3	3	3	3	3	3	2	1	2	3	3	3
	CO4	3	3	3	3	3	2	1	3	3	3	3
	CO5	3	2	3	3	2	3	2	3	3	3	3
Average		3.0	2.8	3.0	3.0	3.0	1.8	1.2	2.4	3.0	2.8	2.6

PROFESSIONAL ELECTIVE COURSES - I

L	T	P	C
3	0	0	3

126CAE01 OBJECT ORIENTED SOFTWARE ENGINEERING

BASIC PREREQUISITES:

- Fundamentals of Programming
- Basics of Software Engineering
- Basic knowledge of Object-Oriented Programming

COURSE OBJECTIVES:

- Understand software development processes, lifecycle models and Agile methodologies.
- Learn object-oriented analysis and design techniques for software system modeling.
- Apply UML, design principles and design patterns in object-oriented software development.
- Understand software testing techniques and maintenance strategies for object-oriented systems.
- Learn software estimation techniques, quality models and object-oriented metrics.

UNIT - I SOFTWARE DEVELOPMENT AND PROCESS MODELS

9

Introduction to Software Development – Challenges – An Engineering Perspective – Object Orientation – Software Development Process – Iterative Development Process – Process Models – Life Cycle Models – Unified Process – Iterative and Incremental – Agile Processes.

Case Study

A software company adopted Agile and iterative development processes to manage changing customer requirements, improve software quality, and deliver applications efficiently within deadlines

UNIT – II AGILE PROCESSES

9

Introduction to Agile Methodology – Need for Agile – Agile vs Traditional Development – Types of Agile Methods – Agile Manifesto and Principles – Agile Values – Agile Team and Collaboration – Agile Project Management – Agile in Software Development and Testing. Introduction to Lean Software Development – Introduction to Scrum – Scrum Roles and Responsibilities – Scrum Events – Scrum Artifacts – Product Backlog and Sprint Backlog – User Stories – Sprint Planning – Daily Scrum – Sprint Review and Retrospective – Agile Estimation – Project Velocity – Burndown Chart – Benefits and Challenges of Agile.

Case Study

A software development team adopted Agile methodology to develop a College Event Management System. The team worked collaboratively, developed the system in small iterations, collected regular feedback, and performed continuous testing. Agile helped the team respond to changing requirements, improve software quality, and deliver the system efficiently.

UNIT - III MODELING OO SYSTEMS

9

Object Oriented Analysis (OOA / Coad-Yourdon)- Object Oriented Design (OOD/Booch)- Hierarchical Object Oriented Design (HOOD)- Object Modeling Technique (OMT) – Requirement Elicitation – Use Cases – SRS Document – OOA - Identification of Classes and Relationships- Identifying State and Behavior – OOD - Interaction Diagrams – Sequence Diagram – Collaboration Diagrams - Unified Modeling Language and Tools.

Case Study

A software company applied design principles, design patterns, and object modeling techniques to develop a scalable and maintainable enterprise application.

UNIT - IV SYSTEM TESTING

9

Software testing: Software Verification Techniques – Object Oriented Checklist :- Functional Testing – Structural Testing – Class Testing – Mutation Testing – Levels of Testing – Static and Dynamic Testing Tools - Software Maintenance – Categories – Challenges of Software Maintenance – Maintenance of Object Oriented Software – Regression Testing.

Case Study

A software maintenance team used functional, structural, and regression testing techniques to identify defects, improve software quality, and ensure reliable performance after system updates.

UNIT - V SOFTWARE QUALITY AND METRICS

9

Need of Object-Oriented Software Estimation – Lorenz and Kidd Estimation – Use Case Points Method – Class Point Method – Object Oriented Function Point – Risk Management – Software Quality Models – Analyzing the Metric Data – Metrics for Measuring Size and Structure – Measuring Software Quality - Object Oriented Metrics.

Case Study

A software company used object-oriented estimation methods, software metrics, and quality models to estimate project cost, manage risks, and improve software quality and performance.

Total No. of Periods : 45

COURSE OUTCOMES:

At the end of course- the Student will be able to:

- CO1:** Explain different software development process models, including Unified Process and Agile processes.
- CO2:** Analyze and model software systems using object-oriented analysis and UML techniques.
- CO3:** Apply object-oriented design principles, GRASP principles and GoF design patterns to solve software design problems.
- CO4:** Apply software testing techniques and maintenance strategies to ensure the reliability and maintainability of object-oriented software.
- CO5:** Apply software estimation methods, quality models and object-oriented metrics to evaluate software projects and product quality.

REFERENCES

1. Roger S. Pressman- “Software Engineering: A Practitioner’s Approach- Tata McGraw-Hill Education- 8th Edition- 2015
2. Yogesh Singh- RuchikaMalhotra- “ Object – Oriented Software Engineering”- PHI Learning Private Limited -First edition-2012
3. Ivar Jacobson. Magnus Christerson- PatrikJonsson- Gunnar Overgaard- “Object Oriented Software Engineering- A Use Case Driven Approach”- Pearson Education- Seventh Impression- 2009
4. Craig Larman- “Applying UML and Patterns- an Introduction to Object-Oriented Analysis and Design and Iterative Development”- Pearson Education- Third Edition- 2008
5. Watts S. Humphrey- “A Discipline for Software Engineering”- Pearson Education- Edition 2007.
6. James F Peters and Witold Pedrycz-” Software Engineering-An Engineering Approach: John Wiley and Sons- New Delhi- 2009.
7. Shari Lawrence Pfleeger- Joanne M. Atlee- “Software Engineering: Theory and Practice”- Fourth Edition- Pearson Education- 2010.
8. Ian Sommerville- “Software Engineering”- Ninth Edition- Pearson Education- 2011

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE01 OBJECT ORIENTED SOFTWARE ENGINEERING	CO1	3	2	2	2	3	2	2	2	2	3	2
	CO2	3	3	3	3	2	2	2	2	3	3	2
	CO3	3	3	3	3	2	2	2	2	3	3	2
	CO4	3	2	3	3	2	2	2	2	3	2	2
	CO5	3	2	3	2	2	3	2	2	3	2	2
Average		3.00	2.40	2.80	2.60	2.20	2.20	2.00	2.00	2.80	2.60	2.00

126CAE02 COMPUTER NETWORKS

BASIC PREREQUISITES:

- Knowledge of Types of Networks
- Functionalities of Various Topology
- Process of Packets

COURSE OBJECTIVES :

- To learn the fundamental concept of OSI Layer- Networks and Protocol standards.
- To understand about Wired and Wireless Network in Data Link Layer
- To Explore the concept of Network Protocol and Routing
- To Know the basic concept of Transport Layer
- To understand basic skill in Application Layer.

UNIT - I DATA COMMUNICATION

9

Data Communication – Networks - Protocols and Standards – Network Models – Layers in the OSI model – TCP/IP Protocol Suite – Addressing – Transmission Modes – Transmission Media Telephone Network – Dial-Up Modems – Digital Subscriber Line – Cable TV Networks – Cable TV for Data Transfer.

Case Study

A broadband service provider used networking models, TCP/IP protocols, transmission media, and cable TV networks to deliver reliable internet and communication services to customers.

UNIT - II DATA LINK LAYER

9

Error Detection and Error Correction : Block Coding – Linear Block Codes – Cyclic Codes – Checksum – Data Link Control : Framing – Flow and error control – Protocols – Noiseless channels – Noisy channels – HDLC – Point to point protocols – Wired LANs: Ethernet – IEEE standard - Standard Ethernet – Changes in the standard – Fast Ethernet – Gigabit Ethernet – Wireless LANs: IEEE 802.11 – Bluetooth – SONET/SDH – Architecture – SONET layers – SONET frames – STS Multiplexing – SONET networks – Virtual Tributaries.

Case Study

A communication network company used error detection, Ethernet standards, wireless LAN technologies, and SONET architecture to ensure reliable high-speed data transmission across wired and wireless networks.

UNIT - III NETWORK LAYER

9

Internetworking – Ipv4 Addresses – Ipv6 Addresses -Ipv4 – Ipv6 – Transition from IPv4 to IPv6– Address Mapping –ICMP – IGMP – ICMPv6 – Delivery- Forwarding – Routing – Unicast Routing protocols – Multicast Routing protocols

Case Study

An internet service provider implemented IPv4, IPv6, routing protocols, and address mapping techniques to improve efficient data delivery, network connectivity, and communication across large networks

UNIT - IV TRANSPORT LAYER**9**

Process-to-Process Delivery – User Datagram Protocol – TCP – SCTP – Data Traffic – Congestion and its Control – Quality of Service – Improve QoS – Integrated Services – QoS in Switched Networks.

Case Study

A cloud networking company used TCP, UDP, SCTP, congestion control, and Quality of Service techniques to manage data traffic and ensure reliable communication in high-speed networks.

UNIT - V APPLICATION LAYER**9**

Domain Name System – Remote Logging – Electronic Mail – File Transfer – WWW and HTTP Simple Network Management Protocol – Cryptography – Symmetric-Key and Asymmetric- Key Cryptography – Security Service – Message Confidentiality- Integrity and Authentication –IPSec – SSL – PGP – Firewall.

Case Study

A banking organization implemented DNS, HTTP, cryptography, SSL, IPSec, PGP, and firewall technologies to provide secure online transactions and protect customer data from cyber threats

Total Number of Periods: 45**COURSE OUTCOMES:**

At the end of course- the Student will be able to:

- CO1 :** Understand principles of data communication- network protocols- and standards- including the OSI and TCP/IP models- various transmission modes- media- and modern telecommunication technologies
- CO2 :** Implement data link layer mechanisms such as framing- flow control- and error control.
- CO3 :** Analyze and evaluate different routing algorithms and their impact on network performance.
- CO4 :** Applying knowledge of transport layer protocols in practical network applications and optimizing their performance.
- CO5 :** Demonstrate proficiency in understanding- configuring- and securing network services.

REFERENCES :

1. Gerry Howser- "Computer Networks and the Internet: A Hands-On Approach"- Springer Publication- 2019
2. Oliver C. Ibe- "Fundamentals of Data Communication Networks"- Wiley Publication- 2017
3. Behrouz A. Forouzan- "Data Communication and Networking"- 4th Edition- McGraw-Hill- 2007.
4. Behrouz A. Forouzan and Firouz Mosharraf- "Computer Networking: A Top-Down Approach"- McGraw-Hill- 2013.
5. Andrew S. Tannenbaum and David J. Wetherall- "Computer Networks"- Pearson- 2013.
6. William Stallings- "Data and Computer Communications"- 8th Edition- Pearson- 2012.
7. Larry L. Peterson & Bruce S. Davie- "Computer Networks – A systems Approach"- 5th Edition- 2012.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE02 COMPUTER NETWORKS	CO1	3	3	2	3	1	1	2	2	3	2	1
	CO2	3	3	2	3	1	1	2	2	3	2	1
	CO3	3	3	3	3	2	1	2	2	3	2	2
	CO4	3	3	3	3	2	1	2	2	3	2	2
	CO5	3	3	3	3	2	1	2	2	3	2	2
Average		3.00	3.00	2.60	3.00	1.60	1.00	2.00	2.00	3.00	2.00	1.60

126CAE03 - OPERATING SYSTEM CONCEPTS

COURSE OBJECTIVES

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that include architecture- Mutual exclusion algorithms- deadlock detection algorithms- and agreement protocols
- To know the components and management aspects of concurrency management

UNIT - I OPERATING SYSTEM AND PROCESS MANAGEMENT 9

Introduction - System Components - Operating System Services - System Calls – System Programs - Process Concept - Process Scheduling – Operations on Process - Cooperation Process - Inter-process Communication - CPU Scheduling and Criteria - Scheduling Algorithms - Multiple-Processor Scheduling - Real-Time Scheduling.

Case Study

A real-time operating system used process scheduling, inter-process communication, and CPU scheduling algorithms to efficiently manage multiple tasks and improve system performance.

UNIT - II CRITICAL SECTION AND DEADLOCKS 9

The Critical-Section Problem - Synchronization Hardware – Semaphores - Classic problems of Synchronization - Critical regions – Monitors. System Model – Deadlock Characterization – Methods for Handling Deadlocks – Deadlocks Prevention and Avoidance – Deadlock Detection and Recovery.

Case Study

A banking transaction system used synchronization techniques, semaphores, and deadlock handling methods to manage concurrent transactions safely and avoid resource conflicts.

UNIT - III STORAGE MANAGEMENT 9

Memory Management – Swapping – Contiguous Memory Allocation – Paging – Segmentation– Segmentation with Paging - Virtual Memory – Demand Paging – Process Creation – Page Replacement – Allocation of frames – Thrashing.

Case Study

An operating system used paging, segmentation, virtual memory, and page replacement techniques to efficiently manage memory usage and improve system performance during multitasking

UNIT – IV FILE SYSTEMS INTERFACE AND IMPLEMENTATION 9

File Concept – Access Methods – Directory Structure – File System Mounting – File Sharing – Protection. File System Structure – File System Implementation – Directory Implementation – Allocation Methods – Free-space Management. Kernel I/O Subsystems – Disk Structure – Disk Scheduling – Disk Management – Swap – Space Management.

Case Study

A cloud storage system used file management, disk scheduling, and free-space management techniques to organize data efficiently and improve storage performance and reliability.

UNIT - V WINDOWS ARCHITECTURE

9

Design Goals – System Architecture – System Management Mechanisms – Process and Thread Management – Memory Management – File Systems Management – Input / Output Management – Inter process Communication. Case Study: Linux – Mobile OS

Case Study

Operating systems such as Linux and mobile operating systems use process management, memory management, file systems, and inter-process communication techniques to efficiently manage hardware resources and support multitasking applications.

Total No. of Periods: 45

COURSE OUTCOMES:

- CO1 :** Understand the principles of operating system design and process management.
- CO2 :** Analyze and design the applications to run in parallel either using process or thread models of different OS
- CO3 :** Design- implement- and optimize memory management systems and virtual memory subsystems
- CO4 :** Demonstrate proficiency in file and disk management concepts in operating systems.
- CO5 :** Explore concepts such as virtual memory- demand paging- memory mapping- and memory compaction

REFERENCES :

1. Abraham Silberschatz- Greg Gagne- Peter B. Galvin- "Operating System Concepts"- Eight Edition- Wiley Publication- 2019
2. Ekta. Walia- Ekta- "Operating System Concepts"- Khanna Book Publishing Company (P) Limited-2015
3. Harvey M. Deitel- Paul J. Deitel- and David R. Choffnes- "Operating System"- Third Edition- Pearson- 2013.
4. William Stalling- "Operating Systems Internals and Design Principles"- Fifth Edition- Prentice Hall of India- 2005.
5. Andrew S. Tanenbaum- "Modern Operating Systems"- Third Edition- PHI

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE03 OPERATING SYSTEMS CONCEPTS	CO1	3	3	3	3	1	1	1	2	3	2	1
	CO2	3	3	3	3	2	1	1	2	3	2	1
	CO3	3	3	3	3	2	1	1	2	3	2	2
	CO4	3	3	3	3	2	1	1	2	3	2	2
	CO5	3	3	3	3	2	1	1	2	3	2	2
Average		3.00	3.00	3.00	3.00	1.80	1.00	1.00	2.00	3.00	2.00	1.60

126CAE04 – CYBER SECURITY

BASIC PREREQUISITES:

- Understanding networking concepts like IP addresses- TCP/IP- DNS- and firewalls is essential for grasping how data is secured and transmitted.
- Experience with both Windows and Unix/Linux systems is important- as cybersecurity often involves securing and troubleshooting these platforms.
- Knowing fundamental programming concepts and languages (such as Python) can help in understanding vulnerabilities- scripting- and automating tasks.

OBJECTIVES

- To learn the principles of cyber security and to identify threats and risks.
- To learn how to secure physical assets and develop system security controls.
- To understand how to apply security for Business applications and Network Communications.
- To learn the technical means to achieve security.
- To learn to monitor and audit security measures

UNIT-I PLANNING FOR CYBER SECURITY

9

Best Practices-Standards and a plan of Action-Security Governance Principles- components and Approach-Information Risk Management-Asset Identification-Threat Identification-Vulnerability Identification-Risk Assessment Approaches-Likelihood and Impact Assessment-Risk Tentative Determination- Evaluation and Treatment-Security Management Function-Security Policy Acceptable Use Policy-Security Management Best Practices - Security Models: Bell La Padula model- Biba Integrity Model -Chinese Wall model

Case Study

An organization implemented security governance policies, risk assessment methods, and security models such as Bell-LaPadula and Biba to protect sensitive information and ensure secure system operations.

UNIT-II SECURITY CONTROLS

9

People Management-Human Resource Security-Security Awareness and Education-Information Management- Information Classification and Handling-Privacy-Documents and Record Management-Physical Asset Management-Office Equipment-Industrial Control Systems-Mobile Device Security- System Development-Incorporating Security into SDLC - Disaster management and Incident response planning.

Case Study

An organization implemented information security policies, mobile device security, secure SDLC practices, and disaster recovery planning to protect critical data and ensure business continuity during security incidents.

UNIT-III CYBER SECURITY FOR BUSINESS APPLICATIONS

9

Business Application Management-Corporate Business Application Security-End user Developed Applications-System Access- Authentication Mechanisms-Access Control-System Management Virtual Servers-Network Storage Systems-Network Management Concepts-Firewall-IP Security Electronic Communications - Case study on OWASP vulnerabilities using OWASP ZAP tool.

Case Study

An enterprise organization used authentication mechanisms, access control, firewalls, and the OWASP ZAP tool to identify and fix OWASP web application vulnerabilities and improve overall system security.

UNIT-IV TECHNICAL SECURITY**9**

Supply Chain Management-Cloud Security-Security Architecture-Malware Protection-Intrusion Detection-Digital Rights Management-Cryptographic Techniques-Threat and Incident Management-Vulnerability Management-Security Event Management-Forensic Investigations Local Environment Management-Business Continuity.

Case Study

A cloud service company implemented malware protection, intrusion detection, cryptographic techniques, and business continuity planning to secure its supply chain operations and manage cybersecurity threats effectively.

UNIT-V SECURITY ISSUES**9**

Security Monitoring and Improvement-Security Audit-Security Performance-Information Risk Reporting-Information Security Compliance Monitoring-Security Monitoring and Improvement Best Practices.

Case Study

An organization implemented security auditing, compliance monitoring, and risk reporting practices to continuously monitor system security and improve overall information security performance.

Total No. of Periods: 45**COURSE OUTCOMES**

At the end of the Course- the Student will be able to

- CO1 :** Understanding of the organization's cybersecurity frameworks- policies- and practices- ensuring foundational knowledge of security principles and practices.
- CO2 :** Apply appropriate management- operational- and technical controls to actively safeguard against potential threats and vulnerabilities in the organization's systems.
- CO3 :** Analyze audit results and performance data from cybersecurity controls to assess their effectiveness and identify any potential weaknesses or gaps.
- CO4 :** Evaluate identified security gaps within the system- considering the impact and likelihood of each gap- and prioritize remediation efforts to enhance overall security.
- CO5 :** Create detailed reports on vulnerabilities and proposed improvements- including actionable plans to address and mitigate identified security risks.

REFERENCES :

1. William Stallings- “Effective Cyber Security - A guide to using Best Practices and Standards”- Addison-Wesley Professional- First Edition- 2019.
2. Adam Shostack- “Threat Modelling - Designing for Security”- Wiley Publications- First Edition- 2014.
3. Gregory J. Touhill and C. Joseph Touhill- “Cyber Security for Executives – A Practical Guide”- Wiley Publications- First Edition- 2014.
4. Raef Meeuwisse- “Cyber Security for Beginners”- Second Edition- Cyber Simplicity Ltd- 2017.
5. Patrick Engebretson- “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”- 2nd Edition- Syngress- 2013.
6. Charles P. Pfleeger- Shari Lawrence Pfleeger- Jonathan Margulies- “Security in Computing”- Fifth Edition- Prentice Hall- 2015.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE04 CYBER SECURITY	CO1	3	2	2	2	1	1	2	2	3	2	1
	CO2	2	3	2	3	2	2	3	2	3	2	2
	CO3	2	3	2	2	1	2	2	2	2	2	1
	CO4	2	3	3	2	2	2	2	2	3	2	2
	CO5	2	2	3	2	3	3	2	2	3	3	2
Average		2.2	2.6	2.4	2.2	1.8	2.0	2.2	2.0	2.8	2.2	1.6

126CAE05 ARTIFICIAL INTELLIGENCE

BASIC PREREQUISITES:

- Basic knowledge of Programming
- Fundamentals of Data Structures and Algorithms
- Basic knowledge of Probability and Logic

COURSE OBJECTIVES :

- To understand the fundamental concepts, goals and techniques of Artificial Intelligence.
- To learn various knowledge representation methods for solving AI problems.
- To apply reasoning techniques for solving problems under certainty and uncertainty.
- To understand the fundamental concepts and techniques of Natural Language Processing.
- To analyze and apply parallel and distributed Artificial Intelligence techniques for solving complex computational problems.

UNIT - I INTRODUCTION

9

Artificial Intelligence Definition-Goal of Artificial Intelligence-AI technique- Progress of AI - Uninformed Search -General State space search- Depth First Search-Depth Limited Search-Iterative Deepening Search-Breadth First Search-Uniform Cost Search-Informed Search-Heuristic Search Techniques-Generate and Test-Best First Search-Hill Climbing

Case Study

An intelligent navigation system used uninformed and heuristic search algorithms such as BFS, DFS, Best First Search, and Hill Climbing to find optimal paths and improve decision-making efficiency.

UNIT - II KNOWLEDGE REPRESENTATION

9

Knowledge Representation-Representation and Mappings-Approaches to Knowledge Representation-Propositional Calculus- Predicate Calculus-Clausal Form-Inference Rules-Unification Algorithm-Resolution

Case Study

An expert system used knowledge representation techniques, predicate logic, and inference rules to analyze information and provide intelligent decision-making and problem-solving support

UNIT – III REASONING UNDER UNCERTAINTY

9

Non monotonic Reasoning-Bayes Theorem-Certainty Factors-Bayesian Networks-Dempster-Shaffer Theory-Fuzzy Logic—Semantic Nets-Frames-Conceptual Dependency-Scripts-CYC-Game Playing

Case Study

An intelligent medical diagnosis system used Bayesian networks, fuzzy logic, semantic networks, and certainty factors to analyze uncertain patient data and support accurate clinical decision-making.

UNIT - IV NATURAL LANGUAGE PROCESSING

9

Introduction to Natural Language Processing – Syntactic Processing – Semantic Analysis – Discourse and Pragmatic Processing – Stochastic Tools for Language Analysis – Natural Language Applications – Psychological Modeling – Language Models – Information Retrieval – Information Extraction – Machine Translation – Speech Recognition.

Case Study

An intelligent chatbot system used natural language processing, syntactic and semantic analysis, language models, and information retrieval techniques to understand user queries and provide accurate responses.

UNIT - V PARALLEL AND DISTRIBUTED ARTIFICIAL INTELLIGENCE

9

Parallelism in Reasoning Systems – Distributed Reasoning Systems – Parallel and Distributed AI Architectures – Parallel Search Techniques – Distributed Problem Solving – Multi-Agent Systems – Distributed Knowledge Representation – Parallel Knowledge Processing – PROLOG

Case Study

An intelligent AI system used parallel and distributed reasoning techniques to solve complex problems efficiently by dividing computational tasks among multiple processing units and coordinating the results.

Total No. of Periods : 45

COURSE OUTCOMES :

At the end of course- the Student will be able to:

- CO1:** Apply uninformed and heuristic-based search algorithms to solve AI search problems.
- CO2:** Apply knowledge representation and logical inference techniques to represent and reason about AI problems.
- CO3:** Analyze reasoning techniques for solving problems involving uncertainty using probabilistic and fuzzy approaches.
- CO4:** Apply Natural Language Processing techniques including syntactic, semantic, discourse and pragmatic analysis.
- CO5:** Analyze and apply parallel and distributed Artificial Intelligence techniques for solving complex computational problems.

REFERENCES

1. Elaine Rich and Kevin Knight S.B. Nair- “Artificial Intelligence” Tata McGraw-Hill Pub- Delhi- 2009.
2. M. Tim Jones- “Artificial Intelligence” A systems Approach- Firewall Media- Edition 2010.
3. George F Luger- “Artificial Intelligence- structures and strategies for complex problem solving”- Pearson Education- Delhi- Edition 2001
4. Sharbani Bhattacharya-” Artificial Intelligence”- University Science Press- New Delhi- 2008.
5. S. Russell and P. Norvig- "Artificial Intelligence: A Modern Approach"- Prentice Hall- Third Edition- 2009.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE05 ARTIFICIAL INTELLIGENCE	CO1	3	3	3	3	2	1	1	3	3	2	2
	CO2	3	3	3	3	2	1	1	3	3	2	2
	CO3	3	3	3	3	2	1	1	3	3	2	2
	CO4	3	3	3	3	2	1	1	3	3	2	2
	CO5	3	3	3	3	2	1	1	3	3	2	2
Average		3.0	3.0	3.0	3.0	2.0	1.0	1.0	3.0	3.0	2.0	2.0

PROFESSIONAL ELECTIVE LAB – I COURSES

L	T	P	C
0	0	4	2

126CAL01 OPERATING SYSTEMS CONCEPTS LAB

BASIC PREREQUISITES:

- Knowledge of Programming in C
- Understanding of Pointers and Functions in C
- Basic knowledge of Data Structures
- Basic understanding of Computer Organization and Operating System concepts

COURSE OBJECTIVES:

- To understand and implement various CPU scheduling algorithms.
- To develop practical skills in UNIX/Linux system calls and file operations.
- To implement process synchronization and deadlock avoidance techniques.
- To understand and implement inter-process communication mechanisms.
- To implement memory management, page replacement, disk scheduling, and file allocation techniques.

LIST OF EXPERIMENTS

1. **CPU Scheduling Algorithms**
FCFS, SJF, Round Robin, Priority
2. **UNIX/Linux I/O System Calls**
Open, read, write, close, fcntl, lseek, stat, opendir, readdir
3. **Banker's Algorithm for Deadlock Avoidance**
4. **Producer–Consumer Problem using Semaphores**
5. **Inter-Process Communication**
Pipes, FIFOs, Message Queues, Shared Memory
6. **Memory Management Techniques**
Paging and Segmentation
7. **Page Replacement Algorithms**
FIFO, LRU, Optimal
8. **Disk Scheduling Algorithms**
FCFS, SSTF, SCAN, C-SCAN
9. **Memory Allocation Techniques**
First Fit, Best Fit, Worst Fit
10. **File Allocation Techniques**
Sequential, Linked, Indexed

Total No. of Periods : 60

COURSE OUTCOMES:

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

- CO1 :** Implement and analyze various CPU scheduling algorithms and UNIX/Linux system calls. (K3)
- CO2 :** Apply deadlock avoidance and process synchronization techniques to solve operating system problems. (K3)
- CO3 :** Implement and demonstrate various inter-process communication mechanisms. (K3)
- CO4 :** Implement and analyze memory management and page replacement techniques. (K4)
- CO5 :** Implement and analyze disk scheduling and file allocation techniques. (K4)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAL01 OPERATING SYSTEMS CONCEPTS LAB	CO1	3	3	3	3	2	1	1	2	3	2	1
	CO2	3	3	3	3	2	1	1	2	3	2	2
	CO3	3	3	3	3	3	1	1	2	3	3	2
	CO4	3	3	3	3	3	1	1	2	3	3	2
	CO5	3	3	3	3	3	1	1	2	3	3	2
Average		3.0	3.0	3.0	3.0	2.6	1.0	1.0	2.0	3.0	2.6	1.8

126CAL02 NETWORK PROGRAMMING LAB

BASIC PREREQUISITES:

- Basic Linux Command
- Concept of Protocols like TCP/IP, UDP
- Structure of Packets.

COURSE OBJECTIVES :

- Understands the fundamental concepts of computer networking and OSI Reference model.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Learn and understand the advanced networking concepts, preparing the student for entry advanced courses in computer networking.
- Develop and gain expertise in some specific areas of networking such as the design and maintenance of individual networks.
- To study about Client-Server models, Processes, Semaphores and their programming.

LIST OF EXPERIMENTS

1. Implement the following forms of IPC.
 - a) Pipes
 - b) FIFO
2. Implement file transfer using Message Queue form of IPC
3. Write a programme to Create an integer variable using shared memory concept and increment the variable
4. Simultaneously by two Processes. Use Semaphores to avoid race conditions
5. Design TCP iterative Client and Server application to reverse the given input sentence
6. Design TCP Client and Server application to transfer file
7. Design a TCP concurrent Server to convert a given text into uppercase using Multiplexing system call “select”
8. Design a TCP concurrent Server to echo given set of sentences using poll functions
9. Design UDP Client and Server application to reverse the given input sentence.
10. Design UDP Client server to transfer a file
11. Design using poll Client-Server application to multiplex TCP and UDP requests for converting a given text into upper case.
12. Design a RPC application to add and subtract a given pair of integers.

Total No. of Periods : 60

COURSE OUTCOMES :

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

- CO1 :** Demonstrate proficiency in implementing Pipes and FIFOs for inter-process communication (IPC) (K3)
- CO2 :** Learn to manage shared memory variables among multiple processes and implement semaphores to ensure synchronized access.(K4)
- CO3 :** Design TCP client-server applications for tasks like text manipulation, file transfer, and concurrent server operations using advanced network programming techniques. (K5)
- CO4 :** Design and implement a UDP Client-Server application (K5)
- CO5 :** Develop network security and define various protocols such as FTP, HTTP, Telnet, DNS (K6)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAL02 NETWORK PROGRAMMING LAB	CO1	3	3	3	3	2	1	1	3	3	2	2
	CO2	3	3	3	3	2	1	1	3	3	2	2
	CO3	3	3	3	3	2	1	1	3	3	2	2
	CO4	3	3	3	3	2	1	1	3	3	2	2
	CO5	3	3	3	3	2	1	1	3	3	2	2
Average		3.0	3.0	3.0	3.0	2.0	1.0	1.0	3.0	3.0	2.0	2.0

126CAL03 SOFTWARE TESTING LAB

BASIC PREREQUISITES:

- Knowledge in Scripting and database.

COURSE OBJECTIVES:

- Develop test plans, test scenarios and test cases for software applications.
- Apply verification, validation and black-box testing techniques to identify software defects.
- Implement unit testing using assertions and exception-handling techniques.
- Perform functional, integration, system, regression, acceptance and usability testing.
- Use test automation and defect management tools to execute tests and prepare test reports.

LIST OF EXPERIMENTS

- 1 Prepare a Test Plan, Test Scenarios and Test Cases for a given software application.
- 2 Perform Verification and Validation of a software application using requirements, design and functional testing.
- 3 Design and execute test cases using Equivalence Partitioning and Boundary Value Analysis.
- 4 Design and execute test cases using Decision Table and State Transition Testing.
- 5 Develop unit test cases using a testing framework and validate results using assertions.
- 6 Develop test cases to verify normal and exceptional conditions in an application.
- 7 Perform Functional, Integration and System Testing for a web-based application.
- 8 Perform Regression, Acceptance and Usability Testing for a given application.
- 9 Automate test cases for a web application using Selenium or a suitable automation tool.
- 10 Execute automated tests, record defects and prepare a Test Execution and Summary Report.

Total No. of Periods : 60

COURSE OUTCOMES :

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

- CO1 :** Apply fundamental software testing principles to prepare test plans, scenarios and test cases. (K3)
- CO2 :** Apply systematic test design techniques for verifying and validating software applications. (K3)
- CO3 :** Develop and execute unit tests using assertions and exception-handling techniques. (K3)
- CO4 :** Perform and analyze different levels and types of software testing to evaluate software quality. (K4)
- CO5 :** Use and evaluate test automation and defect management tools to execute tests and generate test reports. (K5)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAL03 SOFTWARE TESTING LAB	CO1	3	3	3	3	2	1	1	3	3	2	2
	CO2	3	3	3	3	2	1	1	3	3	2	2
	CO3	3	3	3	3	2	1	1	3	3	2	2
	CO4	3	3	3	3	2	1	1	3	3	2	2
	CO5	3	3	3	3	2	1	1	3	3	2	2
Average		3.0	3.0	3.0	3.0	2.0	1.0	1.0	3.0	3.0	2.0	2.0

126CAL04 - CYBER SECURITY LAB

BASIC PREREQUISITES:

- Basic knowledge of Computer Networks
- Basic knowledge of Operating Systems

COURSE OBJECTIVES :

- To understand cybersecurity planning and risk management concepts.
- To learn security controls and incident handling techniques.
- To understand authentication, access control, and web security.
- To learn technical security mechanisms and threat management.
- To understand security auditing and monitoring practices.

LIST OF EXPERIMENTS

1. Study and identify system vulnerabilities using OpenVAS.
2. Configure basic user access permissions in Linux.
3. Configure password and mobile device security settings.
4. Perform data backup and recovery procedures.
5. Configure user authentication and access control.
6. Detect web vulnerabilities using OWASP ZAP tool.
7. Configure firewall settings using pfSense.
8. Perform file encryption and decryption using OpenSSL.
9. Monitor network traffic using Wireshark.
10. Generate basic security audit reports using Wazuh.

COURSE OUTCOMES :

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

- CO1 :** Apply basic cyber security planning and risk management techniques. (K3)
- CO2 :** Implement security controls and incident handling methods. (K3)
- CO3 :** Configure authentication and web security mechanisms. (K3)
- CO4 :** Apply technical security and threat management techniques. (K3)
- CO5 :** Perform security auditing and monitoring for information systems. (K4)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAL04 CYBER SECURITY LAB	CO1	3	3	3	3	2	2	1	2	3	2	2
	CO2	3	3	3	3	2	2	1	2	3	3	2
	CO3	3	3	3	3	3	2	1	2	3	3	3
	CO4	3	3	3	3	3	2	1	2	3	3	3
	CO5	3	3	3	3	3	3	2	3	3	3	3
Average		3.0	3.0	3.0	3.0	2.6	2.2	1.2	2.2	3.0	2.8	2.6

126CAL05 - ARTIFICIAL INTELLIGENCE LAB

BASIC PREREQUISITES:

- Basic knowledge of Programming
- Basic understanding of Data Structures and Algorithms

COURSE OBJECTIVES:

- To understand AI problem-solving and search techniques.
- To learn knowledge representation and reasoning methods.
- To apply reasoning techniques under uncertainty using probabilistic and fuzzy approaches.
- To develop natural language processing applications using AI tools and libraries.
- To implement basic parallel and distributed AI techniques for solving computational problems.

LIST OF EXPERIMENTS

1. Implement Breadth First Search and Depth First Search algorithms.
2. Implement Hill Climbing and Best First Search algorithms.
3. Develop simple propositional and predicate logic programs.
4. Implement Unification and Resolution algorithms.
5. Implement Bayes' Theorem and Fuzzy Logic techniques for handling uncertainty.
6. Develop a simple game-playing program using AI concepts.
7. Implement basic Natural Language Processing tasks using Python and NLTK.
8. Develop a simple chatbot application using Natural Language Processing techniques.
9. Implement speech recognition and text processing applications.
10. Implement a simple parallel or distributed AI problem-solving application using suitable programming tools or frameworks.

COURSE OUTCOMES :

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

- CO1:** Apply AI search algorithms such as Breadth First Search, Depth First Search, Hill Climbing and Best First Search to solve problem-solving tasks. (K3)
- CO2:** Implement knowledge representation techniques, unification and resolution methods for logical reasoning applications. (K3)
- CO3:** Apply probabilistic reasoning and fuzzy logic techniques to handle uncertainty in intelligent systems. (K3)
- CO4:** Develop Natural Language Processing, chatbot and speech processing applications using AI tools and libraries. (K3)
- CO5:** Analyze and implement parallel and distributed AI techniques for solving complex computational problems. (K4)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAL05 ARTIFICIAL INTELLIGENCE LAB	CO1	3	3	3	3	3	1	1	2	3	3	2
	CO2	3	3	3	3	3	1	1	2	3	3	2
	CO3	3	3	3	3	3	1	1	2	3	3	3
	CO4	3	3	3	3	3	2	1	2	3	3	3
	CO5	3	3	3	3	3	2	1	3	3	3	3
Average		3.0	3.0	3.0	3.0	3.0	1.4	1.0	2.2	3.0	3.0	2.6

BRIDGE COURSE

L	T	P	C
2	0	0	0

126CAB01 - FUNDAMENTALS OF COMPUTING AND PROGRAMMING -I

COURSE OBJECTIVES :

- Understand the fundamentals of computers, including their evolution, classification, and basic organization.
- Learn essential data structures and algorithms, with a focus on their real-world applications.
- Gain knowledge of operating systems, including process management and scheduling.
- Explore software engineering principles, life cycle models, and modern development methodologies.
- Develop proficiency in C programming, focusing on core concepts and program development

MODULE – I INTRODUCTION TO COMPUTERS 6

Introduction – Characteristics of Computer – Evolution of Computer – Computer Generation – Classification of Computers – Basic Computer Organization – Number Systems.

MODULE – II DATA STRUCTURES AND ALGORITHMS 6

Introduction – Abstract Data Types – Stack ADT – Application of Stack – Queue ADT – Application of Queue – Linked List – Doubly Linked List – Circular Linked List – Application of Linked List.

MODULE – III OPERATING SYSTEMS 6

Getting Connected to Internet Applications- Operating System Components – System Services– System calls – System Programs – Process Concepts – Process Scheduling – Operation on Process- Cooperation Process – Inter Process communications – CPU Scheduling and Criteria – Scheduling algorithm.

MODULE –IV SOFTWARE ENGINEERING 6

Introduction-Nature of Software-Software Engineering- Importance of Software Engineering- Software Engineering Life Cycle-Software Process-Software Myths, Generic Process Model- Prescriptive Process model: The Waterfall-Incremental-Evolutionary-Concurrent Models- Specialized Process Models-Component based Developments-Formal Methods Model-Aspect Oriented Software Development-The Unified Process-Agile Software Development Model

MODULE – V INTRODUCTION TO C PROGRAMMING 6

Overview of C Programming – Constants, Variable and Data Types – Operators and Expression Managing Input and Output Operators – Decision Making - Branching and Looping - Handling of Character String – User Defined Functions – Definition – Declarations – Call by Reference – Call by Value- Structure and Unions – Pointers - Arrays – The Preprocessor – Developing C Program

Total Number of Hours : 30

COURSE OUTCOMES :

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

- CO1 :** Explain the characteristics, classification, and basic organization of computers, along with number systems. (K2)
- CO2 :** Implement and apply abstract data types like stacks, queues, and linked lists to solve problems. (K3)
- CO3 :** Understand operating system concepts, including process scheduling and inter-process communication. (K2)
- CO4 :** Apply software engineering methodologies, including waterfall, Agile, and unified models, to software development.(K3)
- CO5 :** Write and develop efficient C programs using data types, control structures, functions, and pointers. (K3)

REFERENCES :

1. Herbert Schildt, "C: The Complete Reference", 4th Edn., 2017
2. Pradip Dey, Manas Ghosh, "Programming in C", Oxford University Press. (2007).
3. Byron Gottfried, "Programming with C", 2nd Edition, (Indian Adapted Edition), TMH publications, (2006).
4. Stephen G.Kochan, "Programming in C", Third Edition, Pearson Education India, (2005).
5. Brian W.Kernighan and Dennis M.Ritchie, "The C Programming Language", Pearson Education Inc., (2005).
6. E.Balagurusamy, "Computing fundamentals and C Programming", Tata McGraw-Hill Publishing Company Limited, (2008).
7. S.Thamarai Selvi and R.Murugan, "C for All", Anuradha Publishers, (2008).
8. Abhraham Silberschatz, Greg Gagne, Peter B Galvin, "Operating System Concepts", Eight Edition, Wiley Publication 2019.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAB01 FUNDAMENTA L OF COMPUTING AND PROGRAMMIN G – I	CO1	3	2	2	2	1	1	1	2	3	2	2
	CO2	3	3	2	2	1	1	1	2	3	3	2
	CO3	3	3	2	2	1	1	1	2	3	3	2
	CO4	3	3	3	3	2	1	1	2	3	3	3
	CO5	3	3	3	2	1	1	1	2	3	3	2
Average		3.0	2.8	2.4	2.2	1.2	1.0	1.0	2.0	3.0	2.8	2.2

126CBP01 - FUNDAMENTALS OF COMPUTING AND PROGRAMMING LAB – I

COURSE OBJECTIVES :

- To understand the fundamental concepts of Operators and Conditions statement in C.
- To familiarize students with the concept of loops and their significance in programming.
- To demonstrate various operations and manipulations that can be performed on arrays, such as sorting and searching
- Deep understanding of functions and pointers in C programming
- Understanding of structures and unions in C programming.

LIST OF EXPERIMENTS

1. Implement arithmetic operators using C.
2. Implement Looping Statements using C
3. Implement Conditional Statements using C.
4. Implement Arrays using C.
5. Implement Functions in C.
6. Implement Structure and Union in C.
7. Implement Pointers in C.
8. Implement Stack Operation and Queue Operation using Array.
9. Write a C Code for Polynomial addition using Linked list.

Total No. of Periods: 30

COURSE OUTCOMES :

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

- CO1 :** Demonstrate proficiency in using addition, subtraction, multiplication, division, and modulus operators in C. (K3)
- CO2 :** Write C programs that utilize loops to execute code iteratively based on specific conditions (K3)
- CO3 :** Develop simple application using Array. (K3)
- CO4 :** Deploy programming concepts such as function pointers and pointer (K4)
- CO5 :** Build and Develop an application with structures and unions.(K4)

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CABP01 FUNDAMENTALS OF COMPUTING AND PROGRAMMING LAB- I	CO1	3	2	2	2	1	1	1	2	3	2	2
	CO2	3	3	2	2	1	1	1	2	3	3	2
	CO3	3	3	3	2	1	1	1	2	3	3	2
	CO4	3	3	3	3	2	1	1	2	3	3	3
	CO5	3	3	3	3	2	1	1	2	3	3	3
Average		3.0	2.8	2.6	2.4	1.2	1.0	1.0	2.0	3.0	2.8	2.4

AUDIT COURSES

L	T	P	C
2	0	0	0

126CAE06 - CONSTITUTION OF INDIA

COURSE OBJECTIVES :

- Understand the concept of Historical Movements, principles, and processes involved in the framing of the Indian Constitution
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional.
- Role and entitlement to civil and economic rights as well as the emergence nation hood in the early years of Indian nationalism.
- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.
- To understand the functioning of Union, State and Local Governments in Indian federal system.

UNIT - I HISTORY OF MAKING OF THE INDIAN CONSTITUTION 5
History, Drafting Committee, (Composition & Working)

UNIT - II PHILOSOPHY OF THE INDIAN CONSTITUTION 5
Preamble, Salient Features

UNIT - III CONTOURS OF CONSTITUTIONAL RIGHTS AND DUTIES 5
Fundamental Rights, Right to Equality, Right to Freedom, Right against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Directive Principles of State Policy, Fundamental Duties.

UNIT - IV ORGANS OF GOVERNANCE 5
Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions.

UNIT - V LOCAL ADMINISTRATION 5
District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO, Municipal Corporation. Panchayati raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO Zila Panchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

UNIT - VI ELECTION COMMISSION 5
Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners - Institute and Bodies for the welfare of SC/ST/OBC and women.

Total Number of Hours : 30

COURSE OUTCOMES :

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

- CO1 :** Analyze the significance of Indian Constitution as the fundamental law of the land.(K4)
- CO2 :** Evaluate the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics. (K5)
- CO3 :** Examine the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India. (K4)
- CO4 :** Assess the circumstances surrounding the foundation of the Congress Socialist Party[CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution. (K5)
- CO5 :** Summarize the passage of the Hindu Code Bill of 1956. (K2)

REFERENCES :

1. The Constitution of India,1950(Bare Act), Government Publication.
2. Dr.S.N.Busi, Dr.B. R.Ambedkar framing of Indian Constitution,1st Edition,2015.
3. M.P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis,2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE06 CONSTITUTION OF INDIA	CO1	3	2	2	2	1	2	3	3	3	2	2
	CO2	3	3	2	2	1	2	3	3	3	3	2
	CO3	3	3	2	2	1	2	3	3	3	3	2
	CO4	3	3	2	2	1	2	3	3	3	3	3
	CO5	3	2	2	1	1	2	2	3	3	2	2
Average		3.0	2.6	2.0	1.8	1.0	2.0	2.8	3.0	3.0	2.6	2.2

126CAE07 - DISASTER MANAGEMENT

COURSE OBJECTIVES :

- Summarize basics of disaster
- Explain a critical understanding of key concepts in disaster risk reduction and humanitarian response.
- Illustrate disaster risk reduction and humanitarian response policy and practice from multiple perspectives.
- Describe an understanding of standards of humanitarian response and practical relevance in specific types of disasters and conflict situations.
- Develop the strengths and weaknesses of disaster management approaches

UNIT - I INTRODUCTION

6

Disaster: Definition, Factors and Significance; Difference between Hazard and Disaster; Natural and Manmade Disasters: Difference, Nature, Types and Magnitude.

UNIT - II REPERCUSSIONS OF DISASTERS AND HAZARDS

6

Economic Damage, Loss of Human and Animal Life, Destruction of Ecosystem. Natural Disasters: Earthquakes, Volcanisms, Cyclones, Tsunamis, Floods, Droughts and Famines, Landslides and Avalanches, Man-made disaster: Nuclear Reactor Meltdown, Industrial Accidents, Oil Slicks and Spills, Outbreaks of Disease and Epidemics, War and Conflicts.

UNIT - III DISASTER PRONE AREAS IN INDIA

6

Study of Seismic Zones; Areas Prone to Floods and Droughts, Landslides and Avalanches; Areas Prone to Cyclonic and Coastal Hazards with Special Reference to Tsunami; Post-Disaster Diseases and Epidemics

UNIT - IV DISASTER PREPAREDNESS AND MANAGEMENT

6

Preparedness: Monitoring of Phenomena Triggering a Disaster or Hazard; Evaluation of Risk: Application of Remote Sensing, Data from Meteorological and Other Agencies, Media Reports: Governmental and Community Preparedness.

UNIT - V RISK ASSESSMENT

6

Disaster Risk: Concept and Elements, Disaster Risk Reduction, Global and National Disaster Risk Situation. Techniques of Risk Assessment, Global Co-Operation in Risk Assessment and Warning, People's Participation in Risk Assessment. Strategies for Survival

Total Number of Hours : 30

COURSE OUTCOMES :

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

- CO1 :** Summarize basics of disaster management (K2)
- CO2 :** Evaluate the key concepts in disaster risk reduction and humanitarian response. (K5)
- CO3 :** Analyze disaster risk reduction and humanitarian response policy and practice from multiple perspectives.(K4)
- CO4 :** Examine the humanitarian response and practical relevance in specific types of disasters and conflict situations.(K4)
- CO5 :** Assess the strengths and weaknesses of disaster management approaches. (K5)

REFERENCES :

1. Goel S. L., Disaster Administration And Management Text And Case Studies”,Deep & Deep Publication Pvt. Ltd., New Delhi,2009.
2. NishithaRai, Singh AK, “Disaster Management in India: Perspectives, issues and strategies”, NewRoyal book Company,2007.
3. Sahni, PardeepEt.Al. ,” Disaster Mitigation Experiences And Reflections”, Prentice Hall of India, New Delhi,2001.

Course Name	COs	Program outcomes (Washington Accord Attributes)								PSOs		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PSO 1	PSO 2	PSO 3
126CAE07 DISASTER MANAGEMENT	CO1	3	3	2	2	1	1	1	2	3	2	1
	CO2	2	3	3	3	2	1	1	2	3	2	2
	CO3	2	2	3	3	2	2	2	2	3	3	2
	CO4	2	3	3	3	2	2	2	3	3	2	2
	CO5	2	2	3	3	3	2	2	3	3	3	3
Average		2.2	2.6	2.8	2.8	2.0	1.6	1.6	2.4	3.0	2.4	2.0