TECHNO INDIA UNIVERSITY

WEST BENGAL

Syllabus

for

4-Years B.Tech.

in

Computer Science and Engineering
(Specialization in Artificial Intelligence)

Academic Year: 2024-2025

Semester 5

Design and Analysis of Algorithm (TIU-UCS-T321)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd., 5th
Course Title: Design and Analysis of Algorithm	**Subject Code:** TIU-UCS-T321
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Understand the Fundamental Principles of Algorithm Design
2. Master Asymptotic Analysis and Notations
3. Analyze Algorithm Efficiency in Different Scenarios
4. Apply Algorithm Design to Real-world Problems.

COURSE OBJECTIVE:

The student will be able to:

CO-1	Explain the basic concepts involved in designing, analyzing, and implementing algorithms.	K2
CO-2	Analyze problem characteristics to devise efficient algorithms tailored to specific tasks.	K3
CO-3	Identify and distinguish between tractable and intractable problems in algorithm design.	K3
CO-4	Utilize algorithm design principles to solve real-world problems, implementing solutions and conducting complexity analyses	K4
CO-5	Assess and compare the efficiency of various algorithms based on time and space complexity.	K3
CO-6	Apply advanced algorithmic techniques, such as approximation and parallel algorithms, to solve complex problems efficiently	K3

COURSE CONTENT:

MODULE 1:	**Foundation of Algorithm & Analysis**	**10 Hours**
Introduction to algorithm design and importance of its analysis, Asymptotic notations and their significance, Complexity analysis of algorithms – best case, worst case and average case with example of Insertion sort, Quick sort and Heap sort, Time & space trade-offs, Analysis of recursive algorithms – Substitution method, Recursion tree method and Masters' theorem, Lower bound for comparison-based sort.		

MODULE 2:	**Algorithmic Paradigms**	**10 Hours**
Classification of algorithm design techniques for problem solving: Brute-force, Divide-and-Conquer, Greedy, Dynamic Programming, Backtracking and Branch-and-Bound, Methodology and application domains, Illustration of the techniques with suitable examples: Activity selection, Huffman code, Knapsack problem, Matrix Chain Multiplication, 8-Queen problem, 15-puzzle problem. [extra problem in tutorial]		
MODULE 3:	**Graph Algorithms**	**12 Hours**
Traversal algorithms: DFS, BFS - concept, complexity analysis and applications, Minimum Spanning Tree finding algorithm: Prim's, Kruskal - concept, complexity analysis, Disjoint set operations, shortest path finding algorithm: single source and all pairs –Bellman-Ford, Dijkstra and Floyd-Warshall, Topological sort, Network flow algorithm: Ford-Fulkerson, Max-flow Min-cut theorem.		
MODULE 4:	**Problem Reducibility and NP-completeness**	**8 Hours**
Problem classification on Computability: P, NP, NP-complete and NP-hard, Reducibility of NP-complete problems with example – Satisfiability, Vertex cover, Traveling Salesman problem, Cook's theorem.		
MODULE 5:	**Advanced Topics**	**5 Hours**
Approximation algorithm, Randomized algorithm technique Amortized analysis.		
	TOTAL LECTURES	**45 Hours**

Books:

1. Introduction to Algorithms- Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein, The MIT Press
2. Fundamentals of computer algorithms by Satroj Sahani and Ellis Horowitz.

Database Management System (TIU-UCS-T301)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd., 5th.
Course Title: Database Management System	**Subject Code:**TIU-UCS-T301
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE :

1. Understand the basic concepts and the applications of database systems and the relational

database design principles.

2. Master the basics of SQL and construct queries using SQL.
3. Familiar with the basic issues of transaction processing and concurrency control and database storage structures and access techniques.

COURSE OUTCOME :

The student will be able to:

CO1:	Understand the core concepts and features of Database Management System	K2
CO2:	Design and development of DBMS solutions based on relevant project work	K3
CO3:	Analyze and troubleshoot database related problems and finding the solution using the DBMS knowledge as acquired	K4
CO4:	Study the latest trends in DBMS and get the connectivity with the cutting-edge technologies	K3
CO5:	Implement database security, backup, and recovery techniques to ensure data integrity.	K3
CO6:	optimize SQL queries and database operations for improved performance.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**2 Hours**
General introduction to database systems, Concept of file System and Disadvantages, Database-DBMS distinction, Role of DBA, Approaches to building a database, Data models, Database management system, Three-schema architecture of a database, Data Independency, Integrity constraints.		
MODULE 2:	**RELATIONAL DATA MODEL**	**2 Hours**
Concept of relations, Schema-instance distinction, Keys, Referential integrity and foreign keys. Relational Algebra Operators: Selection, Projection, Union, Intersection, Set difference, Cross product, Rename, Assignment, Various types of joins, Division, Example queries. Tuple Relational Calculus, Domain Relational Calculus.		
MODULE 3:	**SQL (STRUCTURED QUERY LANGUAGE)**	**7 Hours**
Introduction, Data definition in SQL, Table, key and foreign key definitions, Update behaviors, querying in SQL, Basic select- from- where block and its semantics, Nested queries-correlated and uncorrelated, Notion of aggregation, Aggregation functions group by and having clauses, Embedded SQL		
MODULE 4:	**DATABASE DESIGN CONCEPTS (PART-1)-DEPENDENCIES AND NORMAL FORMS**	**9 Hours**
Importance of a good schema design, Problems encountered with bad schema designs, Motivation for normal forms, dependency theory – functional dependencies, Armstrong's axioms for FD's, Closure of a set of FDs, Minimal covers, Definitions of 1NF, 2NF, 3NF and BCNF, Decompositions and desirable properties of them, Algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, Join dependencies and definition of 5NF, DKNF.		
MODULE 5:	**DATABASE DESIGN CONCEPTS (PART-2) -ER MODEL**	**6 Hours**
Conceptual data modeling-motivation, Entities, Entity types, Various types of attributes,		

Relationships, Relationship types, E/R diagram notation, High-level conceptual modeling, ER Modeling concepts, ER Diagrams, Cardinality constraints Enhanced ER Model: Higher-order relationships, Enhanced ER Model (EER), Weak-entity types, Subclasses and inheritance, Specialization and Generalization, Modeling of UNION types using categories.		
MODULE 6:	**DATA STORAGE AND INDEXES**	**7 Hours**
File organizations, Primary, Secondary index structures, Various index structures - hash-based, Dynamic hashing techniques, Multi-level indexes, B+ trees.		
MODULE 7:	**TRANSACTION PROCESSING AND CONCURRENCY CONTROL**	**9 Hours**
Transaction Fundamentals: OLTP environments, Concurrency issues, need for transactions, Necessary properties of transactions (ACID properties), Transaction states, serializability, Serial schedules, Conflict serializability, View serializability, Recoverable and non-recoverable schedules, Cascading rollbacks, Cascadeless schedules. Concurrency control: Serialized and non-serialized schedules, Testing for serializability, Locking, Lock compatibility matrix, Locking and serializability, Deadlocks and starvation, Two-phase locking (2PL) protocol, Conservative, strict and rigorous 2PL, 2PL with lock conversions, Timestamp-ordering based protocol, Multi versioning protocol, Multi-granularity locking, Deadlock prevention protocols, Wait-die and wound-wait schemes, Time-out based schemes, Deadlock recovery, Nested transactions.		
MODULE 8:	**DATABASE RECOVERY TECHNIQUES**	**3 Hours**
Recovery concepts, Deferred updates technique, Immediate update technique, Shadow paging.		
	TOTAL LECTURES	**45 Hours**

Books:

1. Avi Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts,Tata McGraw - Hill Education.
2. Ramez Elmasri and Shamkant Navathe, Fundamentals of Database Systems, Publisher - Pearson Education, 5th Edition
3. Database systems, 6th edition, Ramez Elmasri, Shamkant B.Navathe, Pearson Education
4. Database Systems Design, Implementation, and Management, Peter Rob & Carlos Coronel, 7th Ed
5. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education
6. Microsoft SQL Server 2019 documentation: Databases - SQL Server | Microsoft Docs
7. Microsoft Azure SQL documentation: Azure SQL documentation - Azure SQL | Microsoft Docs
8. Microsoft Azure CosmosDB documentation: Introduction to Azure Cosmos DB |Microsoft Docs
9. Articles on Microsoft Azure and SQL Server: Sucharita Das, Author at SQLServerCentral

Operating System (TIU-UCS-T317)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd., 5th

Course Title: Operating Systems	**Subject Code:** TIU-UCS-T317
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE :

Enable the student to:

1. Understand the structure, functions, and operations of operating systems, including computing environments and open-source OS.
2. Explore process management, scheduling, multithreading, and inter-process communication with synchronization techniques.
3. Learn memory management strategies, including virtual memory, paging, segmentation, and page replacement techniques.
4. Analyze deadlocks, including detection, prevention, and recovery, along with file system implementation and storage management.
5. Examine system protection, access control mechanisms, security policies, and cryptographic techniques for system security.

COURSE OUTCOME :

The student will be able to:

CO1:	Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.	K2
CO2:	Analyze important algorithms e.g. Process scheduling and memory management algorithms, Disk scheduling algorithms.	K4
CO3:	Categorize the operating system's resource management techniques, dead lock management techniques, memory management techniques.	K4
CO4:	Demonstrate the ability to perform OS tasks in Red Hat Linux Enterprise.	K2
CO5:	Evaluate OS performance through scheduling, memory, and file system optimizations.	K4
CO6:	Develop shell scripts and system programs for process management and automation.	K3

COURSE CONTENT :

MODULE 1:	**INTRODUCTION TO OS**	**8 Hours**
Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems. System Structures: Operating System Services, User and Operating-System Interface, systems call, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.		
MODULE 2:	**PROCESS MANAGEMENT**	**10 Hours**
Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems. Multithreaded Programming: Multithreading models, Thread libraries, Threading issues.		

Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling. Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems – Dining philosophers problem, Readers and writers problem.		
MODULE 3:	**MEMORY MANAGEMENT**	**10 Hours**
Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation. Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.		
MODULE 4:	**DEADLOCKS & FILE SYSTEM**	**9 Hours**
Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention. File Systems: Files, Directories, File system implementation, management and optimization. Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.		
MODULE 5:	**SECURITY, SYSTEM PERFORMANCE**	**8 Hours**
System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights. System Security: Introduction, Program threats, System and network threats, Cryptography for security, User authentication, implementing security defenses, Firewalling to protect systems and networks, Computer security classification. Case Studies: Linux, Microsoft Windows.		
	TOTAL LECTURES	**45 Hours**

Books:

1. Operating System Concepts – Abraham Silberschatz, Peter B. Galvin, Greg Gagne
2. Modern Operating Systems – Andrew S. Tanenbaum, Herbert Bos
3. Operating Systems: Internals and Design Principles – William Stallings
4. Operating Systems: A Concept-Based Approach – Dhananjay M. Dhamdhere

Automata Theory (TIU-UCS-T323)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd, 5th
Course Title: Automata Theory	**Subject Code:** TIU-UCS-T323
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

1. To make the student aware about the basic concepts of different abstract computing methods.

2. To make the student aware about regular languages, regular grammar, regular expression, DFA, NFA, their relationship and closure properties of regular languages,
3. To make the students aware about context free languages(CFL), context free grammar, push down automata, closure properties of CFL, Chomsky normal form(CNF), Greibach normal Form(GNF)
4. To make the student aware about context sensitive grammar

COURSE OUTCOME:

After Completion of the course, the students will be able to:

CO1:	Describe the concepts of formal theory of language, the meaning of computing and algorithms	K2
CO2:	Describe and analyze different models of computing such as FA, CFG/PDA, TM	K4
CO3:	Design above models for problem solving	K3
CO4:	Analyze and identify the strengths and shortcomings of the above computing models	K4
CO5:	Describe basic concepts of complexity theory: solvable and unsolvable problems, complexity classes, etc.	K2
CO6:	Apply formal language and automata theory concepts to real-world computing problems.	K3

COURSE CONTENT:

MODULE 1:	**REGULAR LANGUAGES AND FINITE AUTOMATA**	**15 Hours**
Introduction, Alphabet, Language, and Grammar. Regular Expressions and Languages, Deterministic Finite Automata (DFA) and Equivalence with Regular Expressions, Nondeterministic Finite Automata (NFA) and Equivalence with DFA, Regular Grammars and Equivalence with Finite Automata, Properties of Regular Languages, Pumping Lemma for Regular Languages, Minimization of Finite Automata.		
MODULE 2:	**CONTEXT-FREE GRAMMAR/LANGUAGES**	**16 Hours**
Context-Free Grammars (CFG) and Context-Free Languages (CFL), Production, Parse Tree, and Derivation; Chomsky and Greibach Normal Forms, Non-deterministic Pushdown Automata (PDA) and Equivalence with CFG, Parse Trees, Ambiguity in CFG, Pumping Lemma for Context-Free Languages, Deterministic Pushdown Automata, Closure Properties of CFLs. Chomsky Hierarchy of Languages. Context-Sensitive Grammars: Context-Sensitive Grammars (CSG) and Context sensitive Languages (CSL), Linear Bounded Automata (LBA) and its Equivalence with CSG.		
MODULE 3:	**TURING MACHINES**	**9 Hours**
The Basic Model of Turing Machines (TM), Turing-Recognizable (Recursively Enumerable) and Turing-Decidable (Recursive) Languages and Their Closure Properties, Variants of Turing Machines, Non-deterministic TMs and its Equivalence with Deterministic TMs, Unrestricted		

Grammars and Equivalence with Turing Machines, TMs as Enumerators.		
MODULE 4:	**UNDECIDABILITY**	**5 Hours**
Church-Turing Thesis, Universal Turing Machine, The Universal and Diagonalization Languages, Reduction between Languages and Rice's Theorem, Undecidable Problems about Languages.		
	TOTAL LECTURES	**45 Hours**

Books:

1. John E. Hopcroft, Rajeev Motwani , Jeffrey D. Ullman, Introduction to Automata Theory, Languages, And Computation, Pearson
2. Michael Sipser, Introduction to the Theory of Computation, Cengage
3. Dexter C. Kozen, Automata And Computability, Undergraduate Texts In Computer Science, Springer.
4. John Martin, Introduction To Languages AndThe Theory Of Computation, Tata Mcgraw Hill.
5. Harry R. Lewis And Christos H. Papadimitriou, Elements OfThe Theory Of Computation, Pearson Education Asia.

Image Processing and Pattern Recognition (TIU-UCS-T327)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd, 5th
Course Title: Image Processing and Pattern Recognition	**Subject Code:** TIU-UCS-T327
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:**3

COURSE OBJECTIVE:

1. Introduce students to the fundamental concepts of digital image processing, including sampling, quantization, image transforms, and enhancement techniques.
2. Develop an understanding of spatial and frequency domain methods for image enhancement and their applications.
3. Enable students to apply segmentation techniques such as edge detection, thresholding, and region-based methods for effective image analysis.
4. Provide insights into image representation and description methods, including boundary and regional descriptors.

COURSE OUTCOME:

The students will be able to:

CO1:	Understand the fundamental concepts of digital image processing, including sampling and quantization, image transforms, and image enhancement.	K2
CO2:	Apply spatial and frequency domain methods to enhance images.	K3
CO3:	Segment images using edge detection, thresholding, and region-based	K3

	methods.	
CO4:	Represent and describe images using different schemes.	K4
CO5:	Understand the fundamental problems in pattern recognition, including classification, clustering, and feature selection.	K2
CO6:	Implement and evaluate image processing and pattern recognition techniques in real-world applications.	K3

COURSE CONTENT:

MODULE 1:	**DIGITAL IMAGE FUNDAMENTALS & IMAGE TRANSFORMS**	**5 Hours**
Wavelet Transform Applications in Image Processing, Sampling and Quantization, Binary image Analysis, 2-D FFT, Properties, Walsh Transform, Hadamard Transform, Discrete cosine Transform, Discrete Wavelet Transform.		
MODULE 2:	**IMAGE ENHANCEMENT**	**9 Hours**
Spatial domain methods: Introduction, Image Enhancement in Spatial Domain, Bilateral and Guided Filtering, Enhancement Through Point Operation, Types of Point Operation, Histogram Manipulation, gray level Transformation, local or neighbourhood operation, median filter, spatial domain high- pass filtering. Frequency domain methods: Filtering in Frequency Domain, Obtaining Frequency Domain Filters from Spatial Filters, Generating Filters Directly in the Frequency Domain, Low Pass(smoothing) and High Pass (sharpening) filters in Frequency Domain.		
MODULE 3:	**IMAGE SEGMENTATION AND MORPHOLOGICAL IMAGE PROCESSING**	**7 Hours**
Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region oriented segmentation Dilation and Erosion, structuring element, Opening and closing, The Hit and Miss Transform.		
MODULE 4:	**REPRESENTATION AND DESCRIPTION**	**3 Hours**
Representation schemes, Boundary descriptors, Regional descriptors (Texture, moments)		
MODULE-5:	**FUNDAMENTAL PROBLEM IN PATTERN RECOGNITION**	**5 Hours**
Basic problem of pattern recognition with example, Pattern, Pattern class, Classification, Classifier, Pattern Recognition Model, Feature selection, False alarms.		
MODULE-6:	**CLUSTERING**	**5 Hours**
Fundamental of lustering, Metric and non-metric proximity, Density estimation (Parzen window approach, nearest neighbor approach), Seed point selection (Single seed, Multi seed techniques), Hierarchical clustering (Agglomerative, Divisive: K-means, ISODATA), Fuzzy C-means		
MODULE-7:	**CLASSIFICATION**	**5 Hours**
Pattern classification by likelihood function, Bayes classifier, Artificial Neural Net (Neuron, types of neurons, Neural network model, Hopfield net algorithm, Single layer perceptron algorithm and multi-layer perceptron algorithm)		

MODULE-8:	REMOTE SENSING AND APPLICATION	6 Hours
Characteristics of remote sensing (resolution, bands, spectral range, spectral reflection, LANDSAT, SPOT, IRS -1C), Classification of remote sensing data (Minimum distance classifier, Bayes classifier, parallelepiped classifier, multi-seed technique, Support Vector Machine), Application of remote sensing data.		
	TOTAL LECTURES	**45 Hours**

Books:

1. R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Pearson, 2017, ISBN-10: 0133356728, ISBN-13: 978-0133356724.
2. B. B. Chaudhuri and U. Pal, "Digital Document Processing: Major Directions and Recent Advances", Springer, 2007, ISBN-10: 184628501X, ISBN-13: 978-1846285013.
3. E. R. Davies, "Computer and Machine Vision: Theory, Algorithms, Practicalities", Academic Press, 2018, ISBN-10: 0128092847, ISBN-13: 978-0128092842.
4. S. Theodoridis and K. Koutroumbas, "Pattern Recognition", Academic Press, 2008, ISBN-10: 1597492728, ISBN-13: 978-1597492720.

Database Management System Lab (TIU-UCS-L315)

Program: B.Tech. in CSE-AI	**Year, Semester:** 3rd, 5th.
Course Title: Database Management System Lab	**Subject Code:** TIU-UCS-L315
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand the principles of relational databases and SQL.
2. Apply database operations using SQL to manage and manipulate data effectively.
3. Develop complex queries, stored procedures, and triggers for efficient data handling and automation.

COURSE OUTCOME:

Upon completion of the course, the student will be able to:

CO-1	Understand and apply DDL (Data Definition Language) and DML (Data Manipulation Language) statements to create and modify database structures and data.	K3
CO-2	Perform join operations to retrieve data from multiple tables efficiently.	K3
CO-3	Use aggregate functions to analyze and summarize data within a database.	K4
CO-4	Ensure referential integrity and manage relationships between tables in a	K3

	database.	
CO-5	Implement indexing and views to optimize database performance and retrieval operations.	K4
CO-6	Utilize transactions, stored procedures, and triggers to ensure data integrity and automate database operations.	K3

COURSE CONTENT:

MODULE 1:	**DDL AND DML OPERATIONS**	**9 Hours**
Introduction to SQL; DDL Statements: CREATE, ALTER, DROP; DML Statements: INSERT, UPDATE, DELETE; Constraints and data integrity		
MODULE 2:	**JOIN OPERATIONS**	**6 Hours**
Inner Join, Outer Join (Left, Right, Full); Cross Join, Self Join; Performance considerations in join operations		
MODULE 3:	**BUILT-IN FUNCTIONS AND INTEGRITY CONSTRAINTS**	**6 Hours**
Aggregate Functions (SUM, COUNT, AVG, MAX, MIN); String Functions; Integrity Constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY)		
MODULE 4:	**REFERENTIAL INTEGRITY**	**6 Hours**
Concept of Referential Integrity, Enforcing foreign key relationships, Handling cascading updates and deletes		
MODULE 5:	**INDEXING AND VIEWS**	**6 Hours**
Creating and using indexes, Advantages and limitations of indexes, Creating and managing views		
MODULE 6:	**STORED PROCEDURES, TRANSACTIONS, AND TRIGGERS**	**12 Hours**
Creating and executing stored procedures, Transaction Control (COMMIT, ROLLBACK, SAVEPOINT), Creating and managing triggers		
	TOTAL LAB HOURS	**45 Hours**

Books:

1. Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.). Pearson.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts (7th ed.). McGraw-Hill.
3. Ramakrishnan, R., & Gehrke, J. (2014). Database Management Systems (3rd ed.). McGraw-Hill.
4. Date, C. J. (2019). An Introduction to Database Systems (8th ed.). Pearson.

5. Mullins, C. S. (2012). Database Administration: The Complete Guide to DBA Practices and Procedures (2nd ed.). Addison-Wesley.

Design and Analysis of Algorithms Lab (TIU-UCS-L321)

Program: B.Tech. in CSE-AI	**Year, Semester:** 3rd, 5th.
Course Title: Design and Analysis of Algorithms Lab	**Subject Code:** TIU-UCS-L321
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. To understand the fundamental concepts of algorithm design and analyze their time and space complexities.
2. To apply algorithmic techniques such as divide and conquer, dynamic programming, and greedy methods to solve computational problems.
3. To evaluate the efficiency and correctness of algorithms using mathematical analysis and empirical testing.

COURSE OUTCOME :

On completion of the course, the student will be able:

CO-1	Develop and implement sorting algorithms such as Quick Sort and Merge Sort using the Divide and Conquer approach.	K3
CO-2	Apply dynamic programming techniques to solve optimization problems like the 0-1 Knapsack problem.	K3
CO-3	Employ algorithms such as Dijkstra's for solving single-source shortest path problems in graphs.	K4
CO-4	Analyze and examine algorithms like Floyd-Warshall's for finding the shortest path between all pairs of vertices in a graph.	K3
CO-5	Solve and optimize problems like the Travelling Salesman problem using various algorithmic approaches.	K4
CO-6	Evaluate the time and space complexity of algorithms using Big O notation and assess their performance in solving real-world problems.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO ALGORITHMS AND COMPLEXITY ANALYSIS**	**6 Hours**
Definition and Importance of Algorithms, Performance Analysis: Time and Space Complexity, Asymptotic Notation: Big-O, Big-Theta, and Big-Omega, Empirical and Theoretical Analysis: Experimental evaluation with iterative and recursive algorithms		

MODULE 2:	**SORTING AND SEARCHING ALGORITHMS**	**9 Hours**
Sorting Techniques: Bubble Sort, Insertion Sort, Selection Sort (basic sorting). Merge Sort: Divide and Conquer Strategy. Quick Sort: Partitioning and Randomized Pivot Selection. Heap Sort: Using Max-Heap and Min-Heap. Searching Techniques: Linear Search and Binary Search. Complexity Analysis: Best-case, Worst-case, and Average-case performance comparison.		
MODULE 3:	**GREEDY ALGORITHMS**	**6 Hours**
Greedy Methodology: Characteristics and Applicability. Applications: Fractional Knapsack Problem, Minimum Spanning Tree (MST) using Prim's and Kruskal's Algorithms. Complexity Analysis: Time and Space Complexity of Greedy Algorithms.		
MODULE 4:	**DYNAMIC PROGRAMMING (DP)**	**9 Hours**
Dynamic Programming Paradigm: Optimal Substructure and Overlapping Subproblems. Applications: 0/1 Knapsack Problem, Longest Common Subsequence (LCS), All-pairs shortest path using Floyd-Warshall Algorithm. Complexity Analysis: Time and Space Complexity of DP algorithms.		
MODULE 5:	**GRAPH ALGORITHM**	**6 Hours**
Graph Representation: Adjacency Matrix and Adjacency List. Graph Traversal: Breadth-First Search (BFS), Depth-First Search (DFS). Single-Source Shortest Path: Dijkstra's Algorithm.		
MODULE 6:	**BACKTRACKING AND BRANCH & BOUND**	**9 Hours**
Backtracking Concepts: Solving problems using backtracking. Applications: N-Queens Problem. Branch and Bound: Concept and Applications. Solving Traveling Salesman Problem (TSP). Complexity Analysis: Time complexity of backtracking and branch and bound.		
TOTAL LAB HOURS		**45 Hours**

Books:

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MITPress.
2. J. Kleinberg and E. Tardos,AlgorithmDesign,Addison-Wesley.
3. Harry R. Lewis and Larry Denenberg, Data Structures and their Algorithms, Harper Collins.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
5. E. Horowitz and S. Sahani, Fundamentals of Computer Algorithms, Computer Science Press.

Object-Oriented Systems Lab (TIU-UCS-L319)

Program: B.Tech. in CSE-AI	**Year, Semester:** 3rd, 5th.
Course Title: Object-Oriented Systems Lab	**Subject Code:** TIU-UCS-L319
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. obtain the foundational skills to write, compile, and execute basic Java programs, while exploring the use of data types, variables, arrays, and control structures (decision-making and loop control).

2. implement object-oriented principles such as data abstraction, encapsulation, polymorphism, inheritance, interfaces, and packages, using Java, to solve real-world programming problems effectively.
3. create robust Java programs utilizing exception handling, multi-threading, and applet programming, while focusing on debugging, evaluating program correctness, and ensuring code efficiency and maintainability.

COURSE OUTCOME :

Upon completion of the course, the student will be able to:

CO-1	Demonstrate the ability to write, compile, and execute basic Java programs.	K3
CO-2	Illustrate the use of data types, variables, arrays, and control structures such as decision-making (if, nested if) and loop control (do, while, for).	K3
CO-3	Apply concepts of data abstraction, encapsulation, polymorphism, inheritance, interfaces, and packages to solve problems in Java.	K4
CO-4	Develop Java programs incorporating exception handling and multi-threading mechanisms.	K3
CO-5	Execute applet programs and illustrate their usage.	K4
CO-6	Evaluate and debug Java programs for correctness, performance, and maintainability, ensuring efficient use of resources and adherence to best coding practices.	K3

COURSE CONTENT :

MODULE 1:	**INTRODUCTION TO JAVA PROGRAMMING BASICS**	**9 Hours**
Overview of Java programming language, IDE setup, and compiling Java programs; Understanding the basic structure of a Java program, including classes, methods, and variables; Overview of primitive data types (int, float, char, etc.); Operators: Arithmetic, relational, logical, and assignment operators; Introduction to decision-making statements in Java (if, nested if, switch).		
MODULE 2:	**LOOP CONTROL STRUCTURES AND ARRAYS**	**9 Hours**
Implementing for, while, and do-while loops for repeating code execution; Nested loops for multi-level iteration; string operations such as substring(), length(), charAt(), etc.; creation and manipulation of single-dimensional and multi-dimensional arrays.		
MODULE 3:	**CLASSES, OBJECTS, AND STATIC MEMBERS**	**6 Hours**
Understanding the basic concepts of classes and objects in Java; Using constructors, instance methods, and instance variables; concept of static members; significance of static variables, methods, and static blocks in Java.		
MODULE 4:	**OBJECT-ORIENTED CONCEPTS: INHERITANCE, POLYMORPHISM, AND ABSTRACTION.**	**6 Hours**
Concept of inheritance in Java: extending classes, constructor chaining, method overriding; Types of inheritance: single, multilevel, and hierarchical inheritance; Understanding the use of abstract classes and abstract methods; Use cases for abstract classes in Java.		
MODULE 5:	**PACKAGES, EXCEPTION HANDLING, AND MULTITHREADING**	**6 Hours**
Introduction to Java packages and their role in organizing code; Demonstrating the use of built-in		

packages (e.g., java.util); Basics of exception handling: try, catch, throw, throws, and finally; Creating custom exceptions and handling multiple exceptions.		
MODULE 6:	**APPLET PROGRAMMING AND GUI DEVELOPMENT WITH AWT**	**9 Hours**
Introduction to applet programming: lifecycle methods (init(), start(), stop(), destroy()); Differences between applets and applications; Overview of GUI programming in Java using AWT; Working with basic GUI components: Button, Label, TextField, etc.; Event handling in AWT components.		
	TOTAL LAB HOURS	**45 Hours**

Books:

1. "Java: The Complete Reference" by Herbert Schildt.
2. "Core Java Volume I—Fundamentals" by Cay S. Horstmann.
3. "Head First Java" by Kathy Sierra and Bert Bates.
4. "Effective Java" by Joshua Bloch.

Operating System Lab (TIU-UCS-L317)

Program: B.Tech. in CSE-AI	**Year, Semester:** 3rd, 5th.
Course Title: Operating System Lab	**Subject Code:** TIU-UCS-L317
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. Understand fundamental operating system concepts such as processes, threads, memory management, and inter-process communication (IPC).
2. Implement and analyze core OS functionalities, including scheduling, file management, and synchronization.
3. Gain hands-on experience with Red Hat Enterprise Linux and practical troubleshooting of OS-related issues.

COURSE OUTCOME :

Upon completion of the course, the student will be able to:

CO-1	Explain fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.	K3
CO-2	Analyze important algorithms such as process scheduling, memory management, and disk scheduling algorithms.	K4
CO-3	Categorize the operating system's resource management techniques, deadlock management techniques, and memory management techniques.	K4
CO-4	Demonstrate the ability to perform OS tasks in Red Hat Linux Enterprise.	K3
CO-5	Implement and test operating system concepts like process synchronization, inter-process communication (IPC), and file management in a practical	K4

	environment.	
CO-6	Evaluate and troubleshoot operating system performance, addressing resource allocation, process management, and system stability issues.	K4

COURSE CONTENT:

MODULE 1:	**PROCESS MANAGEMENT AND SCHEDULING**	**9 Hours**
Concept of processes and threads, CPU scheduling algorithms (FCFS, SJF, RR, Priority), Process creation and management in Linux.		
MODULE 2:	**INTER-PROCESS COMMUNICATION AND SYNCHRONIZATION**	**9 Hours**
IPC mechanisms (pipes, message queues, shared memory), Process synchronization, Semaphores, and mutex locks.		
MODULE 3:	**MEMORY MANAGEMENT TECHNIQUES**	**6 Hours**
Paging and segmentation, Virtual memory, Page replacement algorithms (FIFO, LRU, Optimal).		
MODULE 4:	**FILE SYSTEM AND DISK MANAGEMENT**	**6 Hours**
File operations, File allocation methods, Disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN).		
MODULE 5:	**DEADLOCK HANDLING AND RESOURCE ALLOCATION**	**6 Hours**
Deadlock prevention and avoidance, Banker's algorithm, Resource allocation graphs.		
MODULE 6:	**SYSTEM PERFORMANCE AND SECURITY**	**9 Hours**
Monitoring system performance, Troubleshooting OS issues, and Security management in Linux.		
	TOTAL LAB HOURS	**45 Hours**

Books:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. - Operating System Concepts (10th ed.), Wiley
2. Tanenbaum, A. S., & Bos, H. - Modern Operating Systems (4th ed.), Pearson
3. Dhamdhere, D. M. - Operating Systems: A Concept-Based Approach (3rd ed.), McGraw-Hill
4. Mauro, J., & McDougall, R. - Solaris Internals: Core Kernel Architecture, Prentice Hall

Image Processing and Pattern Recognition Lab (TIU-UCS-L327)

Program: B.Tech. in CSE-AI	**Year, Semester:** 3rd, 5th.
Course Title: Image Processing and Pattern Recognition Lab	**Subject Code:**TIU-UCS-L327
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. Introduce students to fundamental image processing techniques such as image acquisition, storage, and representation, enabling them to understand the basic operations involved in handling digital images.
2. Equip students with practical skills in image enhancement and feature extraction by implementing various filtering techniques and pattern recognition methods for improved image analysis.
3. Develop the ability to design and optimize algorithms using human-driven and machine learning-based approaches for solving real-world pattern recognition and image processing challenges.

COURSE OUTCOME :

On completion of the course, the student will be able:

CO-1	Explain the fundamental operations involved in storing images.	K3
CO-2	Develop various filtering procedures to enhance image quality.	K4
CO-3	Identify and differentiate various features used in high-level image processing.	K4
CO-4	Design algorithms using human-driven techniques to solve problems from large data sets.	K4
CO-5	Create algorithms based on machine learning or statistical patterns for real-world data analysis.	K5
CO-6	Evaluate and optimize image processing algorithms for performance, accuracy, and efficiency in real-time applications and large datasets.	K5

COURSE CONTENT :

MODULE 1:	**FUNDAMENTALS OF IMAGE PROCESSING**	**6 Hours**
Introduction to digital images and pixel representation, Reading and storing images using Python (OpenCV, PIL, NumPy), Basic image transformations: Translation, Rotation, Scaling,Calculation of Centroid, Area, and Perimeter of objects, Understanding image formats and data structures		
MODULE 2:	**IMAGE CONVERSION AND HISTOGRAM OPERATIONS**	**6 Hours**
Image Conversion Techniques: RGB to Grayscale and other color model transformations, Image padding techniques; Histogram Processing: Histogram Equalization and Stretching, Exponential and Logarithmic Operators for contrast enhancement, Applications in medical imaging and remote sensing		
MODULE 3:	**IMAGE FILTERING AND NOISE REDUCTION**	**6 Hours**
Filtering Techniques for Image Enhancement: Mean, Median, and Gaussian filters Noise Reduction Techniques: Salt-and-Pepper Noise Removal, Smoothing and sharpening filters		
MODULE 4:	**EDGE DETECTION AND THRESHOLDING TECHNIQUES**	**9 Hours**
Edge Detection Methods: Sobel, Prewitt, Laplacian operators Thresholding and Segmentation: Global and adaptive thresholding techniques, Binary and multi-level segmentation		

MODULE 5:	FEATURE EXTRACTION FOR IMAGE ANALYSIS	9 Hours
Shape-Based Feature Extraction: Bounding Box and Optimal Bounding Box, Circular and Elliptical Fit for 2D shapes; Texture Feature Extraction: Entropy, Contrast, Energy, Correlation, Applications in industrial inspection and medical diagnostics		
MODULE 6:	**MACHINE LEARNING AND CLASSIFICATION IN IMAGE PROCESSING**	**9 Hours**
Clustering Techniques for Pattern Recognition: K-Means, Fuzzy C-Means, Agglomerative Clustering; Remote Sensing and Supervised Classification: Generating training sets for water, concrete, and vegetation, Applying Minimum Distance and Parallelepiped Classifiers Evaluating classifier performance on real-world datasets		
TOTAL LAB HOURS		**45 Hours**

Books:

1. 1. Frank Y. Shih, Image Processing and Pattern Recognition: Fundamentals and Techniques, Wiley, 2010.
2. Rafael C. Gonza Lez, Richard E. Woods, Digital Image Processing,Fourth Edition, Pearson.

Introduction to Data Science (TIU-UCS-S303B)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd, 5th
Course Title: Introduction to Data Science	**Subject Code:** TIU-UCS-S303B
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE :

Enable the student to:

1. Understand data science fundamentals, lifecycle, and big data.
2. Use Python for data collection, preprocessing, and visualization.
3. Apply ML techniques and statistical methods for analysis.

COURSE OUTCOME :

The student will be able to:

CO-1:	Explain data science fundamentals, its lifecycle, and the role of big data.	K2
CO-2:	Utilize Python and key libraries for data collection, preprocessing, and visualization.	K3

CO-3:	Apply statistical methods to interpret data and identify patterns and trends.	K3
CO-4:	Analyze machine learning techniques and their applicability to various problems.	K4
CO-5:	Design and implement a data science project integrating analysis and ML techniques.	K3
CO-6:	Assess ethical, privacy, and security concerns in data science applications.	K4

COURSE CONTENT :

MODULE 1:	**INTRODUCTION**	**7 Hours**
Concepts and Fundamentals, Definition and importance of Data Science, Difference between Analysis and Reporting, Data Science vs. Business Intelligence. Big Data and Applications, Overview of Big Data, Key applications, and industry relevance. Data Science Lifecycle, Steps in the data science workflow, including problem formulation, data collection, processing, model building, and evaluation. Ethics and Privacy in Data Science, Data privacy, bias in algorithms, ethical considerations in data science.		
MODULE 2:	**Data Collection and Preprocessing**	**7 Hours**
Data Collection, Introduction to data sources, web scraping techniques, and API integration. Data Cleaning and Transformation, Handling missing data, data imputation, outlier treatment, encoding categorical variables. Data Preprocessing Techniques, Rescaling, normalization, feature engineering, dimensionality reduction (PCA, LDA).		
MODULE 3:	**Introduction to Programming Tools**	**7 Hours**
Programming Essentials with Python, Introduction to Python for data science. Libraries for Data Science, In-depth usage of libraries like NumPy, Pandas, Matplotlib, and Scikit-learn. Data Visualization, Bar charts, line charts, scatterplots, histograms, box plots, and dashboards. Exploratory Data Analysis (EDA), Techniques for EDA, identifying patterns and relationships in data.		
MODULE 4:	**Mathematical Foundations for Data Science**	**8 Hours**
Linear Algebra, Vectors, matrices, matrix operations, eigenvalues, eigenvectors. Probability and Statistics, Probability basics, conditional probability, Bayes' theorem, distributions, variance, and standard deviation. Statistical Measures, Mean, median, mode, correlation, and causation, Simpson's Paradox. Feature Selection Techniques, TF-IDF, cosine similarity, feature importance.		
MODULE 5:	**Machine Learning Concepts and Techniques**	**8 Hours**
Supervised and Unsupervised Learning, Overview of classification and clustering, key differences. Regression and Classification Models, Linear Regression, Logistic Regression, K-Nearest Neighbors, Naïve Bayes, SVM, Decision Trees, Random Forests. Model Evaluation and Validation, Overfitting, underfitting, train/test splits, cross-validation, confusion matrix. Advanced Concepts, Introduction to Neural Networks, basics of deep learning, time series analysis.		
MODULE 6:	**Advanced Topics and Applications in Data Science**	**8 Hours**
Time Series Analysis, Linear systems analysis, forecasting techniques. Natural Language Processing (NLP), Text preprocessing, sentiment analysis, and topic modeling. Introduction to Big Data Technologies, Overview of Hadoop, Spark, and their relevance in large-scale data science projects. Capstone Project, A project integrating skills learned, with a focus on solving real-world data science problems.		
TOTAL LECTURES		45 **Hours**

Books:

1. J. Grus, “Data Science from Scratch: First Principles with Python”, O’Reilly Media, 2019, ISBN-10: 1492041130, ISBN-13: 978-1492041139.
2. C. O’Neil and R. Schutt, “Doing Data Science: Straight Talk from the Frontline”, O’Reilly Media, 2013, ISBN-10: 1449358659, ISBN-13: 978-1449358655.
3. A. Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, O’Reilly Media, 2019, ISBN-10: 1492032646, ISBN-13: 978-1492032649.
4. S. Srinivasan, “Guide to Big Data Applications”, Springer, 2017, ISBN-10: 3319538177, ISBN-13: 978-3319538174.
5. J. D. Kelleher and B. Tierney, “Data Science”, The MIT Press, 2018, ISBN-10: 0262535432, ISBN-13: 978-0262535434.
6. T. Mitchell, “Machine Learning”, McGraw-Hill, 1997, ISBN-10: 0070428077, ISBN-13: 978-0070428072.
7. V. Granville, “Developing Analytical Talent: Becoming a Data Scientist”, Wiley, 2014, ISBN-10: 1118810082, ISBN-13: 978-1118810088.

Prompt Engineering (TIU-UCS-S303A)

Program: B. Tech. in CSE-AI	**Year, Semester:** 3rd, 5th
Course Title: Prompt Engineering	**Subject Code:** TIU-UCS-S303A
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. to understand the fundamentals of prompt engineering
2. Master the art of crafting effective prompts for various AI models
3. Explore various prompting techniques to achieve desired outcomes from generative AI
4. Understand the ethical concerns and know the ways to mitigate the associated risks

COURSE OUTCOME:

The student will be able to:

CO-1:	Explain the theoretical foundations of prompt engineering and its evolving significance in LLMs	K1
CO-2:	Analyze various prompting techniques and assess their effectiveness using theoretical models.	K3
CO-3:	Investigate advanced prompt structures and their influence on model behavior.	K3
CO-4:	Evaluate LLM performance and limitations across different prompt scenarios.	K4

CO-5:	Identify and mitigate ethical concerns, biases, and constraints in prompt design.	K2
CO-6:	Implement theoretical principles of prompt engineering in practical applications.	K3

COURSE CONTENT:

MODULE 1:	**Foundations of Large Language Models and Prompt Engineering**	**8 Hours**
Historical perspective on NLP and the rise of LLMs. Theoretical understanding of LLM architecture, including Transformer models. Role of tokenization, embeddings, and attention mechanisms in LLMs. Introduction to prompt engineering: theoretical foundation, purpose, and limitations. Exploration of prompt engineering as a bridge between NLP tasks and LLM capability		
MODULE 2:	**Theoretical Models of Prompting and Prompt Typology**	**8 Hours**
Classification and analysis of different types of prompts (direct, indirect, zero-shot, few-shot). Theoretical underpinnings of zero-shot and few-shot learning in LLMs. Prompt-based learning theory: how LLMs interpret and respond to structured input. Analysis of prompt effectiveness through probabilistic and statistical models. The concept of prompt transferability and adaptability across tasks		
MODULE 3:	**Principles of Prompt Design and Evaluation**	**5 Hours**
Theoretical principles for effective prompt construction (clarity, specificity, contextual relevance). Methods to optimize prompt structures for model coherence and reliability. Analytical frameworks for assessing prompt quality and model interpretability. The role of heuristics and biases in human prompt design. Understanding the relationship between prompt variability and output diversity		
MODULE 4:	**The Impact of Prompt Structure on Model Behavior**	**8 Hours**
Theoretical exploration of prompt-induced bias and model behavior manipulation. Analysis of prompt chaining, task decomposition, and control prompts. Understanding model interpretability: How LLMs respond to and process varied prompts. Theories of response consistency, coherence, and fluency in model output. Introduction to reinforcement learning as a method for optimizing prompt structure		
MODULE 5:	**Applications and Domain-Specific Theories of Prompt Engineering**	**5 Hours**
Overview of domain-specific prompt engineering applications: legal, medical, creative industries, etc. Analytical perspectives on prompt adaptability in specialized fields. Limitations of LLMs in domain-specific tasks and ways to overcome these through prompt design. Domain-specific prompt challenges: specificity, jargon, and context adaptation. Review of case studies where prompt engineering contributed to success in critical domains		
MODULE 6:	**Ethical, Philosophical, and Social Implications of Prompt Engineering**	**6 Hours**
Ethical theories and frameworks as applied to AI and LLM-driven prompt engineering. Theoretical discussion on biases in LLM outputs and prompt-related ethical dilemmas. Philosophical questions on language, meaning, and intent in AI-generated content. Regulatory and ethical guidelines for responsible prompt engineering. Future directions in prompt engineering: ethical considerations in an evolving field		
MODULE 7:	**Project-Based Learning with Theoretical Applications**	**5 Hours**
Synthesis of theoretical principles in real-world prompt engineering applications. Project-based exploration of prompt engineering in novel applications. Comparative analysis of theoretical models vs. practical outcomes in prompt engineering. Capstone project focused on domain-specific prompt design, tuning, and evaluation		

TOTAL LECTURES	45 Hours

Books:

1. Nathan Hunter, "The Art of Prompt Engineering with Chatgpt: A Hands-On Guide: 3 (Learn AI Tools the Fun Way!)" 2023, ISBN: 1739296710, ISBN-13: 978-1739296711.
2. James Phoenix, "Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs", Eighth Edition (O'Reilly Media), 2024, ISBN-13: 9781098153434.

Career Advancement & Skill Development-V SAP (TIU-UCS-S303C)

Program: B. Tech. in CSE-AI	**Year, Semester: 3rd**, 5th
Course Title: Career Advancement & Skill Development-V SAP	**Subject Code:** TIU-UCS-S303C
Contact Hours/Week: 2–0–0 (L–T–P)	**Credit:** 2

COURSE OBJECTIVE :

Enable the student to:

1. understand SAP architecture and ABAP programming concepts, including program flow, object navigator, and transaction management.
2. develop proficiency in ABAP programming, covering statements, loops, modularization, OOP, and database handling.
3. enhance data modeling and reporting skills using transparent tables, Open SQL, and ALV grid control.

COURSE OUTCOME :

The student will be able to:

CO-1	Understand SAP System Architecture and program flow.	K2
CO-2	Develop ABAP programs using SAP Object Navigator and Repository.	K3
CO-3	Implement ABAP statements, logical expressions, and loops.	K3
CO-4	Design and manage ABAP structures and transparent tables.	K4
CO-5	Apply object-oriented programming concepts in ABAP.	K3
CO-6	Utilize Open SQL, database handling, and ALV reporting in SAP.	K4

COURSE CONTENT :

MODULE 1:	**SAP SYSTEM ARCHITECTURE & ABAP BASICS**	**6 Hours**
SAP System Architecture, Flow of a Program, SAP Object Navigator, Repository, Creating Packages, Developing ABAP Programs, Creating Transactions, Adding Transactions to Favorites		
MODULE 2:	**ABAP PROGRAMMING FUNDAMENTALS**	**6 Hours**
Basic ABAP Statements, ABAP Structures, Logical Expressions, Conditional Statements, Loops, Search Helps (F4), String Manipulation, Selection Screens (Radio Button, Check Box)		
MODULE 3:	**ABAP OBJECT-ORIENTED PROGRAMMING & DATA HANDLING**	**6 Hours**
Object-Oriented Programming (Classes, Objects, Methods, Interfaces), Creating Structures in ABAP,		

Transparent Tables (Data Modeling, Table Creation, Maintenance, Viewing Data), Database Handling (Open SQL, Modifications, Data Retrieval, SQL JOINs)		
MODULE 4:	**ADVANCED ABAP PROGRAMMING**	**6 Hours**
ABAP Subroutines (Procedures, Modularization, Include Programs), ALV Grid Control (ALV Programming, ALV Report Generation)		
MODULE 5:	**SAP APPLICATION DEVELOPMENT & BEST PRACTICES**	**6 Hours**
Best Practices in ABAP Development, Debugging Techniques, Performance Optimization, Real-world Use Cases, Project-based Learning & Hands-on Practice		
	TOTAL LECTURES	**30 Hours**

Books:

1. Berg, B. O., & Moxon, P. (2009). SAP ABAP Programming for Beginners. SAP Press.
2. Haas, S., & Mathew, B. (2019). ABAP Development for SAP S/4HANA. SAP Press.
3. Hardy, P. (2021). ABAP to the Future. SAP Press.
4. Haeuptle, K. (2020). Clean ABAP: A Style Guide for Developers. SAP Press.
5. Keller, H. (2009). ABAP Programming Guidelines. SAP Press.
6. Keller, H., & Krüger, S. (2007). SAP ABAP Objects. SAP Press.
7. Kogent Learning Solutions Inc. (2011). SAP ABAP Handbook. Tata McGraw-Hill Education.
8. Lloyd, K. (2012). SAP ABAP: Advanced Cookbook. Packt Publishing.
9. McGhee, D. (2014). ALV Reports in SAP. SAP Press.
10. Wood, J. (2015). Object-Oriented Programming with ABAP Objects. SAP Press.