# *Syllabi of*
# *Departmental Courses (DC) Courses*
# *B.Tech III Semester*
# *For batch admitted 2022-23*
# *(Computer Science & Engineering)*
# *Under Flexible Curriculum*

**MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR-474005**

**(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)**

Department of Computer Science and Engineering

# DISCRETE STRUCTURES

## (2150301)

## COURSE OBJECTIVES:

- To perceive the knowledge of basic algebra
- To use logical notation to define fundamental mathematical concepts
- To familiarize predicate & propositional logic
- To know about the graph theory and its application in computer engineering
- To familiarize the discrete numeric function and generating function.

**Unit 1:**

Finite and infinite sets, mathematical induction, Principles of inclusion and exclusion, functions and relations, summations, binary relations, equivalence relations, Congruence Relation and partitions, partial ordering relations and lattices, Pigeonhole principle.

**Unit 2:**

Propositional logic, syntax, semantics of Atf (atomic formula), Wff'(well formed formula's),validity and satisfiability of wff' by Quine's method, Normal and closure form of prepositional calculus.

**Unit 3:**

Basic of Graph Theory as a Discrete Structure, planner graphs, Graph Coloring, multi-graphs and weighted graph, shortest path in weighted graph, Introduction to Eularian paths and circuits, Hamiltonian paths and circuits, Introduction to trees, rooted trees, Path length in rooted trees, spanning trees and cut trees.

**Unit 4:**

Introduction to discrete numeric functions and generating functions, Introduction to recurrence relations, linear recurrence relations with constant coefficients, homogeneous solutions, particular solutions and total solutions.

**Unit 5:**

Introduction to group, subgroups, generations and evaluation of power, cosets and Lagrange's theorem, group codes, isomorphism and automorphism, homomorphism and normal sub groups, ring, integral domain and field.

`

## RECOMMENDED BOOKS:

- J. Tremblay and R. Manohar: Discrete Mathematical Structures with Application to Computer science. ● Narsingh Deo: Graph Theory.
- C.L.Liu: Discrete Mathematics.
- K.H. Rosen: Discrete Mathematics and its Applications
- S. Lipschutz, Discrete Mathematics

# COURSE OUTCOMES:

After completing this course, the students will be able to:

**CO1.** Understand logical notation to define and reason mathematically about the fundamental data types and structures used in computer algorithms and systems.

**CO2.** Outline various mathematical concepts along with their applications.

**CO3.** Implement the applications of various types of graphs to solve real life problem.

**CO4.** Apply the mathematical concepts to solve engineering problems.

**CO5.** Analyze the set theory, prepositional logic, graph theory, discrete numeric function and algebraic structure to examine the real world problem.

**CO6.** Design analytical skill and interpret applications of engineering in real time troubleshooting.

Department of Computer Science and Engineering

# OPERATING SYSTEMS
# (2150302)

## COURSE OBJECTIVES

- Provide basic knowledge of computer operating system structures and functioning.
- Compare several different approaches to memory management, file management and process management
- Understand various problems related to concurrent operations and their solutions.

## Unit- I

**Basics of operating systems:** Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security.

**Process management:** Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues.

## Unit-II

**Process synchronization:** Background, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.

**Deadlock:** System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock.

## Unit-III

**Memory management:** Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

## Unit-IV

**Storage management:** Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, RAID Structure.

## Unit-V

**File system interface:** File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, and Free-Space Management.

**System Protection:** Goals, Principles, Domain of Protection, Access Matrix, Access Control.

**RECOMMENDED BOOKS**


- Operating System Concepts, Silberschatz, Ninth Edition, Willey Publication.
- Operating Systems, Internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.
- Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.


---

**COURSE OUTCOMES**

After the successful completion of this course, the student will be able to:

**CO1.** Outline the basic concept of operating systems
**CO2.** Analyze the working of operating system
**CO3.** Examine the working of various scheduling/allocation approaches
**CO4.** Measure the performance of various scheduling/allocation approaches
**CO5.** Analyze the various operating system problems/issues
**CO6.** Develop the Solution of various operating system problems/issues

---

**MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR-474005**
**(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)**
Department of Computer Science and Engineering

# DATABASE MANAGEMENT SYSTEM
# (2150304)

## COURSE OBJECTIVES

- To understand the fundamental concepts of a database management system.
- To analyses database requirements and determine the entities involved in the system and their relationship to one another.
- To develop the logical design of the database using data modelling concepts &normalization.
- To manipulate a database using SQL commands.

## Unit-I

**Introduction:** DBMS Concepts & Architecture, File processing system, limitation of file processing system, Advantages of Database System, Schemas, Instances, Data Independence, Data dictionary, Functions of DBA, Database languages, Data Models: Hierarchical Data Model, Network Data Model & Relational Data Model, E-R Model, Comparison between Models, Introduction of File organization Techniques.

## Unit-II

**Relational Data Models**:Entities & Attributes, Entity types, Key Attributes, Relationships, Domains, Tuples, types of Attributes, Relations, Characteristics of Relations, Keys, Attributes of Relation, Relational Database, Integrity Constraints.
Relational Algebra: Concept and Relational Algebra operations like Select, Project, Join, Division, Union etc.

## Unit-III

**SQL:**Introduction of SQL, features of SQL, Data Definition & Data Manipulation commands in SQL, SQL operators, Update Statements & Views in SQL, Query & Sub query, Data Retrieval Queries & Data Manipulation Statements examples etc.Overview of Tuple Oriented Calculus & Domain Oriented Relational Calculus.

## Unit-IV

**Normalization:**Introduction to Normalization, concepts of anomalies and its types, closure set of dependencies and of attributes, Various Normal Forms: 1NF, 2NF, 3NF, BCNF, Functional Dependency, Decomposition, Dependency Preservation, Loss Less & Lossy Join, Definition of Dangling Tuple, and Multi-values Dependencies.

## Unit-V

**Transaction Processing & Concurrency Control**: TransactionProcessing Concepts, ACID properties, State Diagram, Types of Transaction, Basic idea of serializability, Concurrency Control, Concurrent operation of Databases, Recovery, Types of Recovery, Basic overview of Distributed Databases System and Relational Database Management System, Concepts of Object-Oriented Database System and its tools.

**RECOMMENDED BOOKS**


- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill, 6th Edition.
- Raghu Ramakrishnan, Johannes Gehrke,"Database Management System", McGraw Hill., 3rd Edition.
- Elmasri & Navathe, "Fundamentals of Database System", Addison-Wesley Publishing, 5th Edition.
- Date C.J, "An Introduction to Database", Addison-Wesley Pub Co, 8th Edition.
- B.C. Desai, "An introduction to Database systems"


## COURSE OUTCOMES

After completion of the course students would be able to:

**CO1**. Define the terminology, features, classifications, and characteristics embodied in database systems.

**CO2**. Identify different issues involved in the design and implementation of database system.

**CO3**. Analyse database schema for a given problem domain.

**CO4**. Justify principles for logical design of databases, including the E-R modeling and Normalization approach.

**CO5**. Apply transaction processing concepts and recovery methods over real time data.

**CO6**. Formulate, using relational algebra and SQL, solutions to a broad range of query Problems.

Department of Computer Science and Engineering

# DESIGN & ANALYSIS OF ALGORITHMS (2150303)

## COURSE OBJECTIVE:

- To introduce the topic of algorithms as a precise mathematical concept.
- To demonstrate the familiarity with major algorithm design paradigms and methods of analysis.
- To design efficient algorithms for common computer engineering problems.
- To enhance the skills using well-known algorithms and data structures for solving real-life problems.

---

**Unit-I**

**Introduction to Computational Model:** RAM model, Algorithms and its importance, Recurrences and Asymptotic Notations, Growth of function, Mathematical Analysis of Non-Recursive and Recursive Algorithm, Review of Sorting & Searching Algorithms, Basic Tree and Graph Concept: Binary Search Trees, Height Balanced Tree, B-Trees and Traversal Techniques.

**Unit-II**

**Divide and Conquer Method:** Introduction and its Examples such as Finding the maximum and minimum, Binary Search, Merge Sort, Quick Sort and Strassen's Matrix Multiplication.

**Unit-III**

**Greedy Method:** Introduction, Characteristics, greedy activity selection. **Minimum Cost Spanning Trees**: Prim's and Kruskal's Algorithm, knapsack Problem, Single Source Shortest Path: Dijkstra's single source shortest path algorithm, Huffman Coding.

**Unit-IV**

**Dynamic Programming:** Introduction, The principle of Optimality, Examples of Dynamic Programming Methods such 0/1 Knapsack, Travelling salesman problem, Floyds All Pairs Shortest Path, Longest Common Subsequence and Reliability Design.

**Unit-V**

**Backtracking:** Concept and its Examples like 4-Queen's Problem, Knapsack problem Hamiltonian Circuit Problem, Graph Coloring Problem etc. **Branch and Bound**: Introduction and its Examples like – Travelling
Salesperson Problem etc. **NP Completeness**: Introduction, Class P and NP, Polynomial Reduction, NP-Hard and NP-Complete problem.

**RECOMMENDED BOOKS:**


- Fundamentals of Computer Algorithms, Horowitz & Sahani, Universities press
- Introduction to Algorithms, Coreman Thomas, Leiserson CE, Rivest RL, PHI.
- Design & Analysis of Computer Algorithms, Ullman, Pearson.
- Algorithm Design, Michael T Goodrich, Robarto Tamassia, Wiley India.


**COURSE OUTCOMES:**

After Completion of this course, the students would be able to:

**CO1:** Tell the basic features of an Algorithms.

**CO2:** Outline major Algorithms and Data Structures.

**CO3:** Apply various algorithmic design paradigms.

**CO4:** Analyze the asymptotic performance of Algorithms.

**CO5**: Compare different design techniques to develop algorithms for computational problems.

**CO6**: Design algorithms using greedy strategy, divide and conquer approach, dynamic programming, backtracking, branch and bound approach.

**MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR-474005**
**(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)**

Department of Computer Science and Engineering

# DESIGN AND ANALYSIS OF ALGORITHM
# 2150303
# List of Programs

1. WAP to implement the following using array as data structure and analyze its time Complexity.
   a. Insertion sort b. Selection sort c. Bubble sort d. Quick sort
   e. Bucket sort f. Radix sort g. Heap sort h. Merge sort
2. WAP to implement Linear and Binary Search and analyze its time complexity.
3. WAP to implement Matrix Chain Multiplication and analyze its time complexity.
4. WAP to implement Longest Common Subsequence Problem and analyze its time Complexity.
5. WAP to implement Optimal Binary Search Tree Problem and analyze its time complexity.
6. WAP to implement Huffman Coding and analyze its time complexity.
7. WAP to implement Dijkstra's Algorithm and analyze its time complexity.
8. WAP to implement Bellman Ford Algorithm and analyze its time complexity.
9. WAP to implement DFS and BFS and analyze their time complexities.
10. WAP to Implement 0/1 knapsack using dynamic programming**.**

---

**COURSE OUTCOMES**

After completion of this course, the students would be able to:

**CO1**. Relate the principles of algorithm design in solving problems.

**CO2**. Demonstrate basic algorithms and different problem solving strategies.

**CO3**. Build creativeness and confidence to solve non-conventional problems.

**CO4**. Analyze running times of algorithms using asymptotic analysis.

**CO5.** Compare various algorithm design approaches for solving real world problems.

**CO6**. Design and implement optimization algorithms in specific applications

Department of Computer Science and Engineering

# SOFTWARE ENGINEERING
# (2150305)

## COURSE OBJECTIVES

- To understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.
- To understand project management and risk management associated with various types of projects.
- To know basics of testing and understanding concept of software quality assurance and software configuration management process.

## Unit-I

**Introduction to Software Engineering:** Definition, software engineering-layered Technology, Software Characteristics and Components, Software model: Software Development of Life Cycle Model (SDLC), The Waterfall Model, Iterative Waterfall Model, Prototyping Model, Spiral Model, RAD Model. Selection criteria of model: Characteristics of Requirements, Status of Development Team, Users participation, Type of Project and Associated Risk.

## Unit - II

**Requirement Engineering:** Definition, Requirement Engineering Activity , Types of Requirement- Functional and Non-functional Requirements, User and System Requirements, Requirement Elicitation Methods, Requirement Analysis Methods, Requirement Documentation (SRS), Requirement Validation, Requirement Management.

## Unit – III

**Design Concept, Principle and Methods:** Design Fundamentals, Design Principles, Effective Modular Design, Design Representations, Architectural design, Procedural design, data Directed design, Real Time Design, Object Oriented Design, Coupling and Cohesion.

## Unit - IV

**Software Metrics, Project Management and Estimation:** Metrics in Process and Project domains, Software Measurement, Software Quality Metrics, Project Management- Basics-People, Product, Process, Project, Estimation- Software Project Estimation, Decomposition Techniques- Function Point Estimation, Line of Code (LOC) based estimation, Empirical Estimation, COCOMO Model, Project Scheduling Techniques.

## Unit – V

**Software Testing**: Definitions, Software Testing Life Cycle (STLC), , Test Case Design, Strategic Approach to Software Testing- Verification & Validation , Strategic issues, Criteria for completion of Testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Black Box Testing Techniques, White Box Testing Techniques, Acceptance Testing.

**RECOMMENDED BOOKS**

- Software Engineering, Sommerville, Pearson.
- Software Engineering: A Practitioner's Approach, Roger S. Pressman, McGraw Hill.
- Software Engineering, K.K. Agrawal & Yogesh Singh, New Age Publication.
- Software Engineering, Rajib Mall, PHI.

**COURSE OUTCOMES**

After completion of the course students would be able to:

**CO1.**Explain the various fundamental concepts of software engineering.

**CO2**. Develop the concepts related to software design & analysis.

**CO3**. Compare the techniques for software project management & estimation.

**CO4**. Choose the appropriate model for real life software project.

**CO5.** Design the software using modern tools and technologies.

**CO6.** Test the software through different approaches.

# LAB
# JAVA PROGRAMMING
# (2150306)

**Unit 1**

Introduction to Java Programming, Introduction to Java and its features, Java Development Kit (JDK) installation and setup, Java development environment (IDE) usage, Java syntax and basic programming concepts, Variables, data types, and operators, Control structures: decision-making and loops

**Unit 2**

Object-Oriented Programming in Java, Object-oriented programming (OOP) concepts: classes, objects, inheritance, polymorphism, and encapsulation, Java classes and objects, Constructors and methods, Inheritance and interfaces, Packages and access control.

**Unit 3**

Exception Handling and File Handling, Exception handling: try-catch blocks, multiple catch clauses, and exception hierarchy, Throwing and catching exceptions, File I/O operations: reading from and writing to files, Working with streams and readers/writers, File handling best practices and error handling

**Unit 4**

Java Collections Framework, Overview of Java Collections Framework (JCF), Lists, Sets, and Maps in JCF, ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, etc., Working with collections: adding, retrieving, updating, and deleting elements, Iterators and iterating over collections

**Unit 5**

Multithreading and Java GUI Programming, Multithreading concepts: threads, synchronization, and inter-thread communication, Creating and managing threads in Java, Thread synchronization and deadlock prevention, Introduction to Java GUI (Graphical User Interface) programming Event-driven programming and handling GUI events, Swing components and layout management

## Reference Books

1. "Java: A Beginner's Guide" by Herbert Schildt (McGraw-Hill Education)
2. "Effective Java" by Joshua Bloch (Addison-Wesley Professional)
3. "Head First Java" by Kathy Sierra and Bert Bates (O'Reilly Media)
4. "Java: The Complete Reference" by Herbert Schildt (McGraw-Hill Education)
5. "Java Concurrency in Practice" by Brian Goetz et al. (Addison-Wesley Professional)

## Course Outcomes

1. Demonstrate proficiency in Java programming syntax, control structures, and data types to develop functional applications.
2. Apply object-oriented programming principles, including inheritance, polymorphism, and encapsulation, to design and implement robust Java applications.
3. Implement exception handling techniques and file input/output operations to ensure program stability and data persistence.
4. Utilize the Java Collections Framework to effectively manage and manipulate data structures, such as lists, sets, and maps.
5. Design and develop multithreaded applications, incorporating synchronization mechanisms to ensure thread safety and efficiency.
6. Create interactive graphical user interfaces (GUI) using Java Swing components, incorporating event-driven programming to enhance user experience.

# JAVA PROGRAMMING
# (2150306)
# LIST OF EXPERIMENTS

Experiment 1: Setting up the Java Development Environment

- Install the Java Development Kit (JDK) and an Integrated Development Environment (IDE).
- Write a simple "Hello, World!" program and execute it.

Experiment 2: Implementing Basic Control Structures

- Write a program that demonstrates the use of if-else statements for decision-making.
- Implement loops (for, while) to iterate over a set of numbers or perform a specific task.

Experiment 3: Creating and Manipulating Objects

- Design a class representing a student with relevant attributes and behaviors.
- Create multiple instances of the class and invoke methods to perform operations on the student objects.

Experiment 4: Inheritance and Polymorphism

- Create a base class and derived classes to showcase inheritance.
- Demonstrate polymorphism by invoking methods overridden in derived classes.

Experiment 5: Exception Handling

- Write a program that throws and catches different types of exceptions.
- Handle exceptions using try-catch blocks to prevent program termination.

Experiment 6: File Handling

- Read data from a text file and display its content.
- Write data to a file and verify the successful write operation.

Experiment 7: Working with Java Collections

- Create a collection (e.g., ArrayList) and perform operations like adding, retrieving, and removing elements.
- Iterate over a collection using iterators and demonstrate different collection classes.

Experiment 8: Multithreading

- Create multiple threads and execute them concurrently.
- Implement synchronization mechanisms to prevent thread interference.

Experiment 9: GUI Application Development

- Design a graphical user interface (GUI) using Swing components.
- Implement event handlers for GUI components, such as buttons or text fields.

Experiment 10: Comprehensive Project

- Design and implement a comprehensive Java project that incorporates concepts covered throughout the syllabus.
- Examples could include creating a student management system or a simple game using GUI elements.

# JAVA PROGRAMMING
# (2150306)
# SKILL BASED MINI PROJECTS

1. Student Management System:
    - Design a console-based application to manage student information.
    - Implement functionalities like adding, deleting, and displaying student records.
2. Library Management System:
    - Create a program to manage library operations, including book borrowing, returning, and searching.
    - Implement data structures to store book records efficiently.
3. Calculator Application:
    - Develop a GUI-based calculator application using Swing components.
    - Implement basic arithmetic operations and handle user input.
4. File Encryption and Decryption:
    - Create a program to encrypt and decrypt files using encryption algorithms.
    - Provide options for the user to select the encryption method and specify the file to encrypt/decrypt.
5. Quiz Application:
    - Develop a quiz application that presents multiple-choice questions to the user.
    - Implement a scoring system and display the result at the end of the quiz.
6. Bank Account Management System:
    - Design a program to manage bank accounts, including features like account creation, deposit, withdrawal, and balance inquiry.
    - Implement object-oriented concepts to model bank accounts and transactions.
7. Contact Management Application:
    - Develop a console-based application to manage contacts.
    - Implement functionalities like adding contacts, searching by name, and displaying contact details.
8. Tic-Tac-Toe Game:
    - Create a GUI-based Tic-Tac-Toe game using Swing components.
    - Implement game logic to handle player turns and determine the winner.
9. Weather Forecast Application:
    - Develop a program that retrieves weather data from an API and displays it to the user.
    - Implement features like displaying current weather, forecast for multiple days, and location-based search.
10. Online Shopping System:
    - Design a simple online shopping system with features like browsing products, adding items to the cart, and placing orders.
    - Implement shopping cart functionality and user authentication.