# Advanced Problem-Solving Techniques in Programming

**Semester: V/III - PEC**

**Total hours: 9 hours (9 x 5 = 45 hours)**

---

## Course Outcomes (COs) :

**CO1:** Understand and apply the basic principles of recursion to interpret program flow and solve simple computational tasks.

**CO2:** Demonstrate the ability to work with one-dimensional arrays using foundational techniques such as access, modification, and basic algorithmic approaches.

**CO3:** Handle two-dimensional arrays effectively by performing essential matrix operations and interpreting structured data representations.

**CO4:** Use bitwise operations and core collection structures to manage data efficiently and understand their practical relevance in programming.

**CO5:** Apply the fundamental ideas of hashing, sets, and maps to organize information, perform efficient lookups, and manage key–value based data.

---

## UNIT I – Recursion & Fundamentals                    (9 Hours)

Introduction to Recursion and its Purpose – Types of Recursion and their Basic Use – Understanding Recursive Flow and Function Calls – Call Stack Behavior during Recursion – Simple Numerical Recursion for Foundational Understanding – Comparison between Recursion and Iteration in Program Logic

---

## UNIT II – Arrays & Basic Techniques                    (9 Hours)

Introduction to Arrays and Data Storage – Methods of Accessing and Managing Array Elements – Basic Operations such as Traversal and Modification – Fundamental Idea of Two Pointer Technique for Simple Scenarios – Sliding Window Concept for Handling Continuous Data Segments – Basic Algorithmic Thinking for Array Based Tasks

---

## UNIT III – Two-Dimensional Arrays & Matrix Concepts                    (9 Hours)

Structure of Two Dimensional Arrays – Representation of Matrix Data in Rows and Columns – Different Ways of Traversing a Matrix – Basic Concepts of Matrix Rotation and Rearrangement – Understanding Diagonal Elements and Symmetric Patterns – Simple Techniques for Handling and Modifying Matrix Data

**UNIT IV – Bitwise Operations & Collections**         **(9 Hours)**

Meaning of Bitwise Operators and Their Uses – Binary Representation of Data for Efficient Computation – Basic Bit Level Operations for Simple Manipulations – Introduction to Collections in Java and Their Role – Foundational Usage of ArrayList and LinkedList – Basic Applications of Stacks and Queues in Program Flow

---

**UNIT V – Hashing, Sets & Maps**         **(9 Hours)**

Meaning and Purpose of Hashing – Understanding How Basic Hash Functions Work – Fundamentals of Set Structures and Their Behavior – Introduction to Map Structures and Key–Value Organization – Basic Ideas of Efficient Data Lookup and Storage – Simple Applications of Hashing in Everyday Programming

---