

Satyug Darshan Institute of Technology, Faridabad

Programme: BCA (Gen/DS)

Year/Semester: 1/1

Subject Name : Problem Solving Techniques

**Subject
Code:BCG-103-V1**

Department: Computer Science and Engineering

AY:2025-2026

Prepared By: Ms. Nikita

Unit -1

Short Questions

S. No.	Questions	Bloom Taxonomy(KL)
1	Define a problem and a problem instance with one suitable example.	KL-1
2	Explain the concepts of generalization and special cases in problem solving.	KL-2
3	Define pre-condition and post-condition with a suitable example.	KL-1
4	Explain input validation and discuss its necessity in problem solving.	KL-2
5	Define algorithm and program.	KL-1
6	State any two parameters used in the analysis of a problem.	KL-1
7	State any two characteristics of a good algorithm.	KL-1
8	Explain the decomposition of a problem in short.	KL-2
9	Explain the role of data structures in problem solving.	KL-2
10	Explain the role of operators in algorithms.	KL-2

Long Questions

1	Explain the classification of problems with suitable examples.	KL-2
2	Describe the steps involved in problem solving: Understand, Plan, Execute, and	KL-2
3	Explain different problem-solving approaches such as Brute Force, Divide and Conquer, Greedy, Backtracking, and Dynamic Programming with practical examples.	KL-2
5	Define a computational problem and discuss its types with suitable examples.	KL-1
6	Define data structure and explain its types using real-life examples.	KL-1
7	Explain the importance of data structures in problem solving.	KL-2
8	Analyze why problem decomposition is essential for solving complex problems	KL-4

Unit - 2

Short Questions

S. No.	Questions	Bloom Taxonomy
1	Define control structure and list its types.	KL-1
2	State the difference between stacking and nesting with suitable examples.	KL-2
3	Define a flowchart and state its advantages.	KL-1
4	Define pseudocode and state its advantages and disadvantages.	KL-1
5	Explain the limitations of flowcharts and discuss when pseudocode is	KL-2
6	List any four rules of good pseudocode.	KL-1
7	Define entry-controlled loops and exit-controlled loops.	KL-1
8	Define a counter-controlled loop and compare it with a sentinel-controlled	KL-1
9	State the difference between ASCII and Unicode.	KL-1
10	Explain typed and typeless languages.	KL-2
11	Define debugging and list some techniques of debugging.	KL-1
12	Explain why incremental compilation is preferred in modular programming.	KL-2

Long Questions

1	Write an algorithm and draw a flowchart to check whether a number is even or odd.	KL-3
2	Explain how different types of data are represented in memory, including integer representation (signed magnitude, 1's complement, 2's complement), real number	KL-2
3	Compare counter-controlled (definite) and sentinel-controlled (indefinite) loops	KL-4
4	Write an algorithm, draw a flowchart, and write a C program to print Fibonacci or	KL-3
5	Write a C program to calculate approximate values of $\sin x$, $\cos x$, or π using Taylor	KL-3
6	Explain different types of loops with suitable examples. (OR) Explain repetition	KL-2
7	Describe control structures with suitable examples.	KL-2
8	Describe the characteristics of a good algorithm.	KL-2
9	Explain the evolution of programming languages with key features and justify why each new generation became necessary despite the existence of earlier generations.	KL-4
10	Justify the statement "Control structures are the backbone of procedural programming" using real-life scenarios and practical examples.	KL-4
11	Explain arithmetic, relational, and logical operators with examples and describe how they work together for multi-conditional decision making.	KL-2
12	Explain incremental compilation and testing, and justify why incremental testing is helpful in debugging large programs.	KL-4
13	Justify the statement "Incremental compilation improves program reliability" with	KL-4

Unit - 3

Short Questions

S. No.	Questions	Bloom Taxonomy
1	Define sentinel-controlled repetition.	KL-1
2	State the use of break and continue in C.	KL-1
3	Write the syntax of the if-else ladder.	KL-1
4	Define switch-case.	KL-1
5	State the purpose of increment and decrement operators.	KL-1
6	State the difference between postfix and prefix increment/decrement operators.	KL-1

Long Questions		
1	Write an algorithm and draw a flowchart to check whether a number is a	KL-3
2	Write a C program to calculate factorial using iteration and recursion.	KL-3
3	Write an algorithm to find the prime factors of a given number.	KL-3
4	Write an algorithm and draw a flowchart to check whether a number is an	KL-3
5	Write an algorithm using sentinel-controlled repetition to find maximum, minimum, sum, and average of numbers.	KL-3
6	Write a C program to check whether two numbers are amicable.	KL-3
7	Write a program to extract digits of a number from left to right and right to left.	KL-3
8	Write a program for base conversion (decimal to binary or binary to decimal).	KL-3
9	Explain else-if ladder and switch-case with suitable examples.	KL-2
10	Explain nested if-else and write a program to find the largest of three numbers	KL-3
11	Compare while and do-while loops with suitable examples.	KL-4
Unit - 4		
Short Questions		
S. No.	Questions	Bloom Taxonomy
1	Define modular programming.	KL-1
2	Define top-down design.	KL-1
3	Define bottom-up design.	KL-1
4	State any two advantages of modular programming.	KL-1
5	Define recursion.	KL-1
6	Define function prototype.	KL-1
7	Define return statement.	KL-1
8	Define parameters and arguments.	KL-1
9	Define a one-dimensional (1D) array with a suitable example.	KL-1
10	Define a two-dimensional (2D) array with a suitable example.	KL-1
11	Define operator precedence.	KL-1
12	Define associativity.	KL-1
13	Define a string.	KL-1
14	Define debugging.	KL-1
Long Questions		
1	Compare Top-Down and Bottom-Up solution approaches based on design methodology, modularity, debugging, reusability, and real-life applications.	KL-4
2	Explain the methods of passing parameters to a function using suitable programs.	KL-2
3	Define recursion, state its advantages and disadvantages, and write a program to calculate factorial of a number using recursion.	KL-3
4	Design a C program to find the sum of two matrices A[3][3] and B[3][3] into	KL-3
5	Design a C program to perform matrix multiplication.	KL-3
6	Define strings in C, explain how they are stored in memory, and write a program using any five predefined string functions.	KL-3
7	Write algorithms to perform linear search and binary search.	KL-3
8	Write a program to find maximum, minimum, sum, and average using arrays.	KL-3
9	Explain operator precedence and associativity with examples. OR Explain different	KL-2