

Paper Code(s): ICT-101T	L : 3
Paper: Programming for Problem Solving	C : 3

Lecture Plan: Programming for Problem Solving

Topics to be covered

S.NO	Content	Total Lectures
First Term		
Unit-I		
1	Structure of a C program; preprocessing, compilation, translation tools, linking, invocation and execution; structured programming; C89, C99, C11 and C23 extensions and usage.	2
2	Basic data types (int, char, float, double, bool and fixed-width integers); declarations and initializations; constants using const and #define; variables; basic input/output statements and type/inter-variable conversions.	2
3	Operators and expressions: arithmetic, relational, logical, bitwise, assignment and conditional operators; increment/decrement operators; type conversions, precedence and associativity; evaluation of expressions.	2
4	Decision and selection statements: if, if-else, else-if, nested if and switch; problem solving through decision-making programs; tracing and dry runs.	2
5	Loops: while, do-while, for, infinite loops and nested loops; early exit and jump statements break, continue, goto and return; algorithm development using flowcharts and pseudocode; structured problem solving.	2
Unit-II		
6	Functions: declaration, definition and invocation; user-defined and built-in functions; argument passing by value; recursion; inline functions and macro functions; modular program design and header files.	2
7	Storage classes: auto, static, extern, register and thread_local; scope and lifetime of variables; one-dimensional arrays—declaration, initialization, traversal, insertion and deletion; passing arrays to functions.	2
8	Two-dimensional and multi-dimensional arrays; memory layout; searching algorithms—linear and binary search; sorting—insertion sort; tracing, comparison and implementation of algorithms.	2
9	Strings: character arrays and variable-length character strings; string input/output using gets, puts, fgets and fputs; standard string manipulation functions including strlen, strcpy, strcat, strcmp, strncpy and related functions.	2
10	Root finding of a positive floating-point number; problem formulation, algorithm, pseudocode, flowchart and C implementation; integrated practice using functions, arrays, searching, sorting and strings.	2
Second Term		
Unit-III		
11	Structures: struct definition, declaration, initialization and member access; nested structures; arrays of structures; pointers to structures; structures and functions; self-referential structures.	2
12	Unions and enumerated types: union definition and use-cases; enum and typedef; bit fields; memory layout and alignment; comparison and appropriate use of structures and unions.	2
13	Pointers: declaration, referencing and dereferencing; pointer arithmetic; pointers to arrays; array of pointers; pointers to pointers; generic/void pointers; pointer and strings; pointers to structures.	2
14	Dynamic memory allocation using malloc, calloc, realloc and free; memory layout; memory-leak prevention; NULL and null-pointer checks; compound literals; pointer to functions; functions returning pointers.	2
15	Polynomial representation and arithmetic using floating-point coefficients and matrices: addition, subtraction and multiplication; file I/O—opening, closing, reading and writing text and binary files using fopen, fclose, fprintf, fscanf, fread, fwrite, fgets and fputs; error handling using errno; command-line parameters.	2
Unit-IV		
16	Abstract Data Types (ADTs); abstraction and interface/implementation concepts; singly linked lists—creation, traversal, insertion and deletion; dynamic memory and pointer safety.	2
17	Doubly linked lists and circular linked lists—creation, traversal, insertion and deletion; implementation using structures and pointers; safe memory handling.	2
18	Stack ADT using arrays and linked lists; push, pop and peek operations; applications of stacks; infix-to-postfix conversion and parenthesis balancing.	2
19	Queue ADT: linear, circular and priority queues; array and linked-list implementations; enqueue, dequeue and priority processing; applications and comparative analysis.	2
20	Binary trees and binary search trees: creation, representation, inorder/preorder/postorder traversal, searching, insertion and deletion; function pointers, pointers to arrays/functions; _Static_assert, _Generic type-generic programming, assertions and diagnostics (assert), safer macros, forward declaration, program segmentation and header hygiene; course integration.	2

Total Lectures: 40