Programming for Problem Solving Module – 1 : (Introduction to Programming) [08] Introduction to components of a computer system: Memory, processor, I/O Devices, storage, operating system, Concept of assembler, compiler, interpreter, loader and linker.

Idea of Algorithm: Representation of Algorithm, Flowchart, Pseudo code with examples, From algorithms to programs, source code. Programming Basics: Structure of C program, writing and executing the first C program, Syntax and logical errors in compilation, object and executable code. Components of C language. Standard I/O in C, Fundamental data types, Variables and memory locations, Storage classes.

Module – 2 : (Arithmetic expressions & Conditional Branching) [08] Arithmetic expressions and precedence:Operators and expression using numeric and relational operators, mixed operands, type conversion, logical operators, bit operations, assignment operator, operator precedence and associativity. Conditional Branching: Applying if and switch statements, nesting if and else, use of break and default with switch.

Module – 3 : (Loops & Functions) [08] Iteration and loops: use of while, do while and for loops, multiple loop variables, use of break and continue statements. Functions: Introduction, types of functions, functions with array, passing parameters to functions, call by value, call by reference, recursive functions.

Module – 4 : (Arrays & Basic Algorithms) [08] Arrays: Array notation and representation, manipulating array elements, using multi dimensional arrays. Character arrays and strings, Structure, union, enumerated data types, Array of structures, Passing arrays to functions. Basic Algorithms: Searching &Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, Notion of order of complexity.

Module – 5 :( Pointer& File Handling) [08] Pointers:Introduction, declaration, applications, Introduction to dynamic memory allocation (malloc, calloc, realloc, free), Use of pointers in self-referential structures, notion of linked list (no implementation) File handling:File I/O functions, Standard C preprocessors, defining and calling macros, command-line arguments. COURSE OUTCOMES 1. To develop simple algorithms for arithmetic and logical problems. 2. To translate the algorithms to programs & execution (in C language). 3. To implement conditional branching, iteration and recursion. 4. To decompose a problem into functions and synthesize a complete program using divide and conquer approach. 5. To use arrays, pointers and structures to develop algorithms and programs.