

Programming for Problem Solving

Unit 1

Introduction to Programming **(4 lectures)**

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) - **(1 lecture)**.

Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. **(1 lecture)**

From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code- **(2 lectures)**

Unit 2

Arithmetic expressions and precedence **(2 lectures)**

Conditional Branching and Loops **(6 lectures)**

Writing and evaluation of conditionals and consequent branching **(3 lectures)**

Iteration and loops **(3 lectures)**

Unit 3

Arrays **(6 lectures)**

Arrays (1-D, 2-D), Character arrays and Strings

Unit 4

Basic Algorithms **(6 lectures)**

Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)

Unit 5

Function **(5 lectures)**

Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference

Unit 6

Recursion **(4 -5 lectures)**

Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

Unit 7

Structure **(4 lectures)**

Structures, Defining structures and Array of Structures

Unit 8

Pointers **(2 lectures)**

Idea of pointers, Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation)

Unit 9

File handling (only if time is available, otherwise should be done as part of the lab)