# Problem Solving and Programming
## (Common to All Branches of Engineering)

B. Tech – I Semester

L-T-P-C
3-1-0-4

**Course Objectives:**
1. Introduce the internal parts of a computer, and peripherals.
2. Introduce the Concept of Algorithm and use it to solve computational problems
3. Identify the computational and non-computational problems
4. Teach the syntax and semantics of a C Programming language
5. Demonstrate the use of Control structures of C Programming language
6. Illustrate the methodology for solving Computational problems

**Outcomes:**
Student should be able to
1. Identify the different peripherals, ports and connecting cables in a PC (L2)
2. Illustrate the working of a Computer (L3)
3. Select the components of a Computer in the market and assemble a computer (L4)
4. Solve complex problems using language independent notations (L3)

## Unit 1:
**Computer Fundamentals:** What is a Computer, Evolution of Computers, Generations of Computers, Classification of Computers, Anatomy of a Computer, Memory revisited, Introduction to Operating systems, Operational overview of a CPU.

**Introduction to Programming, Algorithms and Flowcharts**: Programs and Programming, Programming languages, Compiler, Interpreter, Loader, Linker, Program execution, Fourth generation languages, Fifth generation languages, Classification of Programming languages, Structured programming concept, Algorithms, Pseudo-code, Flowcharts, Strategy for designing algorithms, Tracing an algorithm to depict logic, Specification for converting algorithms into programs.

## Unit 2:
**Introduction to problem solving:** Introduction, the problem-solving aspect, Design and implementation of algorithms – Topdown design, Analysis of Algorithms, the efficiency of algorithms, the analysis of algorithms.

**Fundamental algorithms:** Exchanging the values of two variables, counting, summation of a set of numbers, factorial computation, sine function computation, generation of the Fibonacci sequence, reversing the digits of an integer.

**Learning Outcomes**: Student should be able to
1. Solve Computational problems (L3)
2. Apply Algorithmic approach to solving problems (L3)
3. Analyze the algorithms (L4)

**Unit 3:**

**Types, Operators, and Expressions:** Variable names, data types and sizes, constants, declarations, arithmetic operators, relational and logical operators, type conversions, increment and decrement operators, bitwise operators, assignment operators and expressions, conditional expressions precedence and order of evaluation.

**Input and output:** standard input and output, formatted output-Printf, formatted input-Scanf

**Control Flow:** Statements and blocks, if-else, else-if, switch, Loops-while and for, Loops-Dowhile, break and continue, goto and labels.

**Functions and Program Structure:** Basics of functions, functions returning non-integers, external variables, scope variables, header variables, register variables, block structure, initialization, recursion, the C processor.

**Learning Outcomes:** Student should be able to
1. Recognize the programming elements of C Programming language (L1)
2. Select the control structure for solving the problem (L4)
3. Apply modular approach for solving the problem (L3)

**Unit 4:**

**Factoring methods:** Finding the square root of a number, the smallest divisor of a number, the greatest common divisor of two integers, generating prime numbers.

**Pointers and arrays:** Pointers and addresses, pointers and function arguments, pointers and arrays, address arithmetic, character pointers and functions, pointer array; pointers to pointers, Multi-dimensional arrays, initialization of arrays, pointer vs. multi-dimensional arrays, command line arguments, pointers to functions, complicated declarations.

**Array Techniques:** Array order reversal, finding the maximum number in a set, removal of duplicates from an order array, finding the $k^{th}$ smallest element.

**Learning Outcomes:** Student should be able to
1. Solve mathematical problems using C Programming language (L3)
2. Structure the individual data elements to simplify the solutions (L6)
3. Facilitate efficient memory utilization (L6)

**Unit 5:**

**Sorting and Searching:** Sorting by selection, sorting by exchange, sorting by insertion, sorting by partitioning, binary search.

**Structures:** Basics of structures, structures and functions, arrays of structures, pointers to structures, self-referential structures, table lookup, typedef, unions, bit-fields.

**Some other Features:** Variable-length argument lists, formatted input-Scanf, file access, Error handling-stderr and exit, Line Input and Output, Miscellaneous Functions.

**Learning Outcomes:** Student should be able to
1. Select sorting algorithm based on the type of the data (L4)
2. Organize heterogeneous data (L6)
3. Design a sorting algorithm (L6)

**Text Books**:
1. Brian W. Kernighan, and Dennis M. Ritchie, "The C Programming Language", 2nd Edition, Pearson.
2. R.G. Dromey, "How to Solve it by Computer". 2014, Pearson.
3. Pradip Dey, and Manas Ghosh, "Programming in C", 2018, Oxford University Press.

**Reference Books:**
1. RS Bichkar "Programming with C", 2012, Universities Press.
2. Pelin Aksoy, and Laura Denardis, "Information Technology in Theory", 2017, Cengage
3. Byron Gottfried and Jitender Kumar Chhabra, "Programming with C", 4th Edition, 2019, McGraw Hill Education.

**Course Outcomes:**
1. Construct his own computer using parts (L6).
2. Recognize the importance of programming language independent constructs (L2)
3. Solve computational problems (L3)
4. Select the features of C language appropriate for solving a problem (L4)
5. Design computer programs for real world problems (L6)
6. Organize the data which is more appropriated for solving a problem (L6)