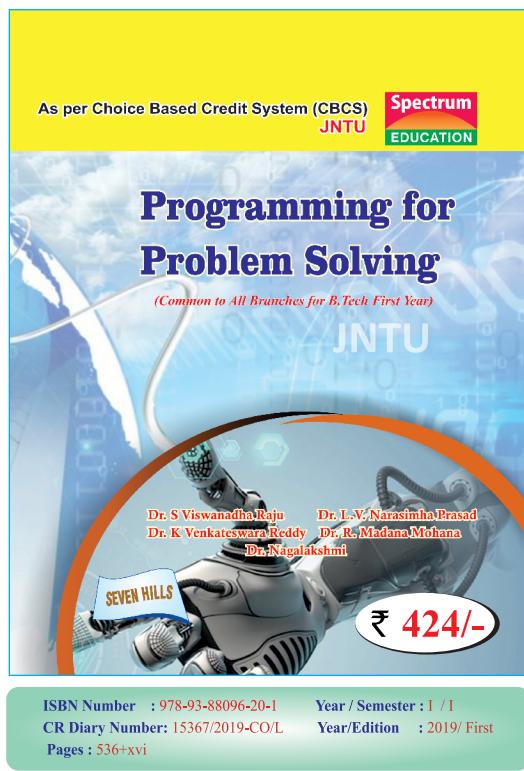


***Programming for Problem Solving*****Chapter-1: Computer and Programming Fundamentals**

- 1.0 Aims and objectives
- 1.1 Introduction of Computers
- 1.2 Classification of Computers
- 1.3 Overview of a Computer
  - 1.3.1 Characteristics
  - 1.3.2 Applications
  - 1.3.3 Limitations
- 1.4 Parts of a Computer
- 1.5 Memory Hierarchy
- 1.6 Introduction to OS
  - 1.6.1 Classification of Operating System
  - 1.6.2 Purpose of Operating System
  - 1.6.3 Functions of Operating System
- 1.7 Operational Overview of a CPU
- 1.8 GenerationS and Classification of Programming Languages
  - 1.8.1 Generations of Computers
  - 1.8.2 Differences between the 3GL and 4GLs
  - 1.8.3 Classification of Programming Languages
  - 1.8.4 Differences between High Level and Low Level Languages/ Machine level languages
- 1.9 Creating and Running Programmes
  - 1.9.1 System Development Tools
    - 1.9.1.1 Comparison of Compiler and Interpreter
  - 1.9.2 Writing, Compiling and Running Your Program

- 1.10 Number System
  - 1.10.1 Binary, Decimal, Hexadecimal And Octal Conversions
- 1.11 Summary
- 1.12 Key Terms
- 1.13 Frequently Asked Questions
- 1.14 Exercises
- 1.15 Multiple choice questions

**Chapter-2: Problem Solving and Algorithms**

- 2.0 Aims and objectives
- 2.1 Introduction to Algorithms
- 2.2 Problem Solving
  - 2.2.1 Steps Involved in Solving Logical and Numerical Problems
- 2.3 Characteristics of an Algorithm
- 2.4 Strategy for Designing Algorithms
- 2.5 Different Ways of Stating Algorithms
  - 2.5.1 Step-form, Pseudo-code, Flowchart
- 2.6 Structured Programming Concept
- 2.7 Implementation of Algorithms
- 2.8 Analysis of Algorithms
- 2.8.1 Efficiency of Algorithm
- 2.9 Simple Examples of the Algorithm
- 2.10 Pseudo Code
  - 2.10.1 Pseudo Code Rules
  - 2.10.2 Advantages of Pseudo Codes
  - 2.10.3 Limitations of Pseudo Codes
- 2.11 Flow Charts
  - 2.11.1 Types of Flow Charts
  - 2.11.2 Advantages of Flowcharts
  - 2.11.3 Differences between Flowchart and Algorithm
  - 2.11.4 Limitations of Flowcharts
  - 2.11.5 Simple Examples of the Flowchart
- 2.12 PRogram Development steps
- 2.13 Tracing an Algorithm to Depict logic
- 2.14 Specification for Converting Algorithms into Programs
- 2.15 Additional ALgorithms and Flowcharts
- 2.16 Summary
- 2.17 Key Terms
- 2.18 Frequently Asked Questions
- 2.19 Exercises
- 2.20 Multiple Choice Questions



### Chapter-3: Basics of 'C'

- 3.0 Aims and objectives
- 3.1 Introduction of 'C'
- 3.2 Characteristics of 'C' Language
- 3.3 'C' Features
- 3.4 'C' Limitations
- 3.5 Basic Structure of 'C' Language
- 3.6 Writing the First C Program, Comments
  - 3.6.1 Compiling and Executing C Programs
  - 3.6.2 Syntax and Logical Errors in Compilation
- 3.7 Program Statements
- 3.8 'C' Tokens
  - 3.8.1 Keywords
  - 3.8.2 Identifiers
  - 3.8.3 Constants
  - 3.8.4 Escape Sequences
  - 3.8.5 Special Symbols
  - 3.8.6 'C' Operators
  - 3.8.7 Variables
    - 3.8.7.1 Declaring Variables
    - 3.8.7.2 Initializing Variables
    - 3.8.7.3 Assigning Values to Variables
- 3.9 Basic Data Types in C
- 3.10 Operators
  - 3.10.1 Arithmetic Operators
  - 3.10.2 Relational Operators
  - 3.10.3 Logical Operators
  - 3.10.4 Assignment Operators
  - 3.10.5 Increment and Decrement Operators
  - 3.10.6 Conditional Operators
  - 3.10.7 Bitwise Operators
  - 3.10.8 Special Operators
- 3.11 Expressions and Evaluation
- 3.12 Precedence and Associativity
- 3.13 Type Conversions
- 3.14 Type Casting
- 3.15 Input and Output Functions
  - 3.15.1 Formatted Input and Output Functions
  - 3.15.2 Non-formatted Input and Output Functions
- 3.16 Additional Programs
- 3.17 Summary
- 3.18 Key Terms
- 3.19 Frequently Asked Questions
- 3.20 Exercise
- 3.21 Multiple Choice Questions

### Chapter - 4: Control Statements

- 4.0 Aims and Objectives
- 4.1 Introduction
- 4.2 Statements
- 4.3 Decision Making (or) Conditional Statements
  - 4.3.1 if, if-else, nested if, nested if-else and else-if Statements
    - 4.3.1.1 if Statement
    - 4.3.1.2 if-else Statement
    - 4.3.1.3 Nested if Statements
    - 4.3.1.4 else-if Statement

- 4.3.1.5 Dangling else Problem
- 4.3.1.6 nested if-else Statements
- 4.3.2 Switch case Statements
- 4.4 Iterative Statements (or) Loop Statements
  - 4.4.1 While Loop/while Statements
  - 4.4.2 Do-while Loop/do-while Statements
  - 4.4.3 For Loop/for Statements
  - 4.4.4 Use of Comma Operator in for Loop
  - 4.4.5 Differences for while, do-while and for loop
- 4.5 Nested Loops
- 4.6 Special Control Statement
  - 4.6.1 Goto Statement
  - 4.6.2 Break Statement
  - 4.6.3 Continue Statement
  - 4.6.4 return Statement
  - 4.6.5 exit Statement
  - 4.6.6 Difference between break and continue Statements
  - 4.6.7 Null Statement
- 4.7 Additional Programs
- 4.8 Factoring Methods
- 4.9 Summary
- 4.10 Key Terms
- 4.11 Frequently Asked Questions
- 4.12 Exercise
- 4.13 Multiple Choice Questions

### Chapter-5: Arrays and Strings

- 5.0 Aims and Objectives
- 5.1 Introduction
- 5.2 Arrays Concepts
  - 5.2.1 Declaration of Arrays
  - 5.2.2 Initialization of Arrays
  - 5.2.3 Accessing Array Elements
  - 5.2.4 Storing Array Elements
- 5.3 Calculating the Length of the Array
- 5.4 Using Arrays in C
  - 5.4.1 Performing Operations on Arrays
  - 5.4.2 Arrays Limitations
- 5.5 Types of Arrays
  - 5.5.1 Onedimensional Arrays
  - 5.5.2 Twodimensional Arrays
  - 5.5.3 Multidimensional Arrays
  - 5.5.4 Comparison of Singledimensional and Multidimensional Arrays
- 5.6 Examples of Two Dimensional Arrays
  - 5.6.1 Addition of Arrays
  - 5.6.2 Subtraction of Arrays
  - 5.6.3 Multiplication of Arrays
- 5.7 Strings
  - 5.7.1 Introduction
  - 5.7.2 Declaration of Strings
- 5.8 String Header or 'C' Library Functions for Strings
- 5.9 Null-terminated String
- 5.10 'C' Strings
- 5.11 Handling Strings as Array of Characters
- 5.12 String Input/Output Functions

- 5.13 Arrays of Strings
- 5.14 String Handling/manipulation Functions
  - 5.14.1 header files in “ctype.h”
- 5.15 Miscellaneous String Functions
- 5.16 Additional Examples
- 5.17 Summary
- 5.18 Key Terms
- 5.19 Frequently Asked Questions

## **Chapter-6: Functions**

- 6.0 Aims and objectives
- 6.1 Introduction
- 6.2 Concept of Function
  - 6.2.1 Purpose of Function
- 6.3 Using Functions
  - 6.3.1 Function Prototype Declaration
  - 6.3.2 Function Definition
  - 6.3.3 Function Calling
  - 6.3.4 Designing Structured Programs
    - 6.3.4.1 Defining and Accessing of Functions
  - 6.3.5 Return Statement
  - 6.3.6 Characteristics of Function
  - 6.3.7 Advantages of Functions
- 6.4 Passing Arguments (or) Passing Parameters
  - 6.4.1 Call-by-Value Vs Call-by-reference
  - 6.4.2 Differences between Call-by-Value and Call-by-reference
  - 6.4.3 Passing Variable Number of Arguments to a Function
- 6.5 Formal Parameters and Actual Parameters
- 6.6 Void Functions
- 6.7 Function Invocation and Function Execution
- 6.8 Nesting of Functions
- 6.9 User defined Functions
- 6.10 Standard/Library functions
  - 6.10.1 Built-in Functions
- 6.11 Static Functions
- 6.12 Scope of Variables
  - 6.12.1 Scope Rules
- 6.13 Storage Classes
  - 6.13.1 Auto Storage Class
  - 6.13.2 Extern Storage Class
  - 6.13.3 Register Storage Class
  - 6.13.4 Static Storage Class
  - 6.13.5 Comparison of Different Storage Class Variables
- 6.14 Passing Arrays to Functions
- 6.15 Type Qualifiers
- 6.16 Inline Functions
- 6.17 Recursion
  - 6.17.1 Recursive Functions
  - 6.17.2 Advantages of Recursion
  - 6.17.3 Limitations of Recursion
  - 6.17.4 Types of Recursion
- 6.18 Towers of Hanoi
- 6.19 Recursion vs Iteration
- 6.20 User defined functions
  - 6.20.1 Introduction

- 6.20.2 Need for User Defined Functions
- 6.20.3 Return Values and Their Types
- 6.20.4 Functions Returning Non-integers
- 6.21 Additional Programs
- 6.22 Summary
- 6.23 Key Terms
- 6.24 Frequently Asked Questions
- 6.25 Exercises
- 6.26 Multiple choice Questions

## **Chapter-7: Pointers**

- 7.0 Aims and objectives
- 7.1 Introduction
- 7.2 Pointers
  - 7.2.1 Declaring Pointer Variables
  - 7.2.2 Assigning Pointers
  - 7.2.3 Initialization of a Pointer
  - 7.2.4 Accessing a Pointer’s Contents
- 7.3 Address and Indirection Operator
- 7.4 Uses of Pointers
- 7.5 Disadvantages of Pointers
- 7.6 Arrays and Pointers
  - 7.6.1 Relationship between Pointers and Arrays
  - 7.6.2 Accessing Elements of Two-Dimensional Array
  - 7.6.3 Pointer and Multidimensional Arrays
- 7.7 Array of Pointers
- 7.8 Dynamic Memory Allocation
- 7.9 Pointers to Pointers
- 7.10 Pointers to Void
- 7.11 Null Pointers
- 7.12 Pointers for inter function communication
- 7.13 Pointers and Strings
- 7.14 Dangling Pointer
- 7.15 Pointer to Array
- 7.16 Function and Pointers
  - 7.16.1 Pointers to Functions
  - 7.16.2 Character Pointers and Functions
- 7.17 Passing Arguments to Functions using Pointer
- 7.18 Difference between Array Name and Pointer
- 7.19 Additional Programs
- 7.20 Summary
- 7.21 Key Terms
- 7.22 Frequently Asked Questions
- 7.23 Exercises
- 7.24 Multiple Choice Questions

## **Chapter-8: Structures and Union**

- 8.0 Aims and objectives
- 8.1 Introduction
- 8.2 Declaring a Structure and its Members
  - 8.2.1 The Type Definition (typedef)
- 8.3 Initialization of a Structure
- 8.4 Accessing Members of a Structure
- 8.5 Assigning Values/Operations on Structures
- 8.6 Size of a Structure
  - 8.6.1 Using sizeof Operator
  - 8.6.2 Without Using sizeof Operator
- 8.7 Array of Structures



- 8.8 Differences between Arrays and Structure
- 8.9 Nested Structures
- 8.10 Structures and Functions
- 8.11 Selfreferential Structures
  - 8.11.1 Usage of Self Referential Structures in Linked List
- 8.12 Table Lookup
- 8.13 Passing Structures through Pointers
- 8.14 Structures Containing Arrays
- 8.15 Structures containing Pointers
- 8.16 Bit Fields
- 8.17 Introduction to Unions
  - 8.17.1 Declaring a Union and its Members
- 8.18 Initialization of a Union
- 8.19 Accessing Members of Union
- 8.20 Arrays of Unions Variables
- 8.21 Unions inside Structures
- 8.22 Enumerated Data Types
- 8.23 Additional Programs
- 8.24 Summary
- 8.25 Key Terms
- 8.26 Frequently Asked Questions
- 8.27 Exercises
- 8.28 Multiple Choice Questions

### **Chapter-9: File Handling in C**

- 9.0 Aims and objectives
- 9.1 Introduction
- 9.2 Using Files C
- 9.3 Types of Files
  - 9.3.1 Differences between Text and Binary Files
- 9.4 file Structure
- 9.5 Streams
- 9.6 File Operations
  - 9.6.1 Opening File
  - 9.6.2 Reading File
  - 9.6.3 Writing to a File
  - 9.6.4 Closing Files
  - 9.6.5 State of File
  - 9.6.6 Appending Data to Existing Files
- 9.7O Verview of Functions
- 9.8 File Input/Output Functions (Standard Library Input/Output Functions for Files)
  - 9.8.1 fread() Function
  - 9.8.2 fwrite() Function
- 9.9 File Handling in C
- 9.10 File Status Functions
  - 9.10.1 Error Handling during File Operations
- 9.11 Positioning Functions
- 9.12 Detecting the End-of-file
- 9.13 Accepting Command Line Arguments
- 9.14 Remove–Renaming a File–Creating a Temporary File

- 9.15 Files of Records, Random Access to Files of Records
- 9.16 Other File Management Functions
- 9.17 Concept of binary files
  - 9.17.1 Reading and Writing from Binary Files
  - 9.17.2 Writing and Reading Structures using Binary Files
- 9.18 Additional Programs
- 9.19 Summary
- 9.20 Exercise
- 9.21 Multiple choice Questions

### **Chapter-10: Searching and Sorting**

- 10.0 aims and objectives
- 10.1 Introduction
- 10.2 Searching
  - 10.2.1 Linear Search
  - 10.2.2 Binary Search
    - 10.2.2.1 Algorithm for Binary Search
    - 10.2.2.2 Algorithm for Binary Search using Recursive Technique
  - 10.2.3 Advantages and Disadvntages of Binary Search
  - 10.2.4 Advantages and Disadvntages of Linear Search
- 10.3 Sorting
  - 10.3.1 Bubble Sort
    - 10.3.1.1 Advantages and Disadvntages of Bubble Sort
  - 10.3.2 Selection Sort
    - 10.3.2.1 Advantages and Disadvntages of Selection Sort
  - 10.3.3 Insertion Sort
    - 10.3.3.1 Advantages and Disadvntages of Selection Sort
  - 10.3.4 Quick Sort
  - 10.3.5 Comparison of Various Sorting Algorithms
- 10.4 Summary
- 10.5 Key Terms
- 10.6 Frequently Asked Questions
- 10.7 Exercises
- 10.8 Multiple Choice Questions

### **Chapter-11: Additional Features of C**

- 11.0 Aims and objectives
- 11.1 Preprocessor Commands
- 11.2 Macros
- 11.3 Command-line Arguments
- 11.4 Variable-length Argument Lists
- 11.5 Error Handling-stderr and Exit
- 11.6 Summary
- 11.7 Exercise