BCS-154 Basics of Programming Skills

|  |  |
| --- | --- |
| Course Category | : PLBSE |
| Pre-requisite Subject | : NIL |
| Contact Hours/Week | : Lecture: 2, Tutorial: 0, Practical: 2 |
| Number of Credits | : 3 |
| Course Assessment  Methods | : Continuous assessment through tutorials, attendance, home assignments,  quizzes, practical work, record, viva voce, one minor test and one  major theory & practical examination |
| Course Outcomes | : The students are expected to be able to demonstrate the following  knowledge, skills, and attitudes after completing this course |

1. Read and understand C programs.

2. Discuss basic theory and practice of programming.

3. Design and implement practical programs using C language.

4. Use compiler and feel comfortable with Windows environment

5. Identify and fix common C errors.

TOPIC COVERED

UNIT-I 9

Basics of Computer: Introduction to Digital Computer, Basic Operations of Computer, Functional Components of Computer, Classification of Computers. Introduction to Operating System: DOS, Windows, Linux, Function, Services and Types. Basics of Programming: Approaches to Problem Solving, Concept of Algorithm and Flow Charts, Types of Computer Languages: Machine Language, Assembly Language and High-Level Language, Concept of Assembler, Compiler, Loader and Linker.

UNIT-II 9

Standard I/O in “C”, Fundamental Data Types: char, int, short, long, float, double, long double. Storage Classes: Automatic, Register, Static, External. Operators and Expressions: Using Numeric and Relational Operators, Mixed Operands and Type Conversion, Logical Operators, Bit Operations, Operator Precedence and Associativity. C Conditional Program Execution: Applying if and Switch Statements, Nesting if and else, Restrictions on switch Values, Use of Break. Program Loops and Iteration: Uses of while, do and for Loops, Multiple Loop Variables, Assignment Operators, Use of break and continue keywords.

UNIT-III 9

Functions: Designing Structured Programs, Functions in C, User Defined and Standard Functions, Formal vs. Actual Arguments, Function Category, Function Prototype, Parameter Passing, Recursive Functions. Arrays: One Dimensional, Multidimensional Array and their Applications, Declaration and Manipulation of Arrays. Strings: String Variable, String Handling Functions, Array of Strings. Storage Classes revisited.

UNIT-IV 9

Pointers: Pointer Variable and its Importance, Pointer Arithmetic and Scale Factor, Compatibility, Dereferencing, L value and R-Value, Pointers and Arrays. Structure and Union: Declaration and Initialization of Structures, Structure and array, Structure Pointers, Declaration and Initialization of union, Union vs Structure. Embedded Devices, Microprocessor, Microcontroller. Implement the concept of Logic Gates, Adders.

EXPERIMENTS

1. Write a program that finds whether a given number is even or odd.

2. Write a program that tells whether a given year is a leap year or not.

3. Write a program that accepts marks of five subjects and finds percentage and prints grades

according to the following criteria:

a. Between 90-100%--------------Print „A‟

b. 80-90%----------------------------Print „B‟

c. 60-80%---------------------------Print „C‟

d. Below 60%----------------------Print „D‟

4. Write a program that takes two operands and one operator from the user and perform the

operation and prints the result by using Switch statement.

5. Write a program to print sum of even and odd numbers from 1 to N numbers.

6. Write a program to print the Fibonacci series.

7. Write a program to check whether the entered number is prime or not.

8. Write a program to find the reverse of a number.

9. Write a program to print Armstrong Numbers from 1 to 100.

10. Write a program to convert binary number into decimal number and vice versa.

11. Write a program that simply takes elements of the array from the user and finds the sum of

these elements.

12. Write a program that inputs two arrays and saves sum of corresponding elements of these

arrays in a third array and prints them.

13. Write a program to find the minimum and maximum element of the array.

14. Write programs to implement the concept of functions and pointer.

15. Write programs to implement the concept of structure and union.

16. Write a program to implement the concept of Logic Gates.

17. Write programs to implement the Adders.

TEXTBOOK

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 7th Edition,

Pearson.

2. Schildt, Herbert, Complete Reference with C, Tata McGraw Hill.

3. Kerninghan and Ritchie, The C programming Language, 2nd Edition, Prentice Hall.