L	T	P	C
3	-	2	4

21BC103 PROGRAMMING FOR PROBLEM SOLVING- I

Course description and objectives:

This course is aimed to impart knowledge on basic concepts of C programming language and problem solving through programming. It covers basic structure of C program, data types, operators, decision making statements, loops, functions, static data structures. At the end of this course students will be able to design, implement, test and debug modular C programs.

Course Outcomes:

The student will be able to:

- Learn about fundamentals of computer and programming language, draw flow chart to solve given problem logically and develop algorithm to solve given program.
- Comprehend the general structure of C program, concepts of variable, datatype, operator and be able to create a C program to demonstrates these concepts.
- Use the concept of branching and looping to design efficient C program and be able to apply the concepts of user defined function and recursion to support reusability.
- Design an application using the concepts of array, pointer, structure and file management to solve real world problem.

Skills:

- Analysis of the problem to be solved.
- Design of algorithm/solution for a given problem.
- Identification of suitable data types for operands.
- Application of suitable control statements for decision making.

Activities:

- Design of non-recursive and recursive functions to perform different tasks.
- Selection of static or dynamic data structures for a given problem and manipulation of data items.
- Development of C programs that are understandable, debuggable, maintainable and more likely to work correctly in the first attempt.

Syllabus

UNIT - I **12Hours**

INTRODUCTION TO ALGORITHMS AND PROGRAMMING LANGUAGES: Basics of algorithms, Flowcharts, Generations of programming languages. **Introduction to C:** Structure of a C program; pre-processor statement, Inline comments, Variable declaration statement, Executable statement; C Tokens: C Character set, Identifiers and Keywords, Type Qualifiers and Type Modifiers, Variables and Constants, Punctuations, and Operators.

UNIT - II **12Hours**

DATA TYPES AND OPERATORS: Basic data types; Storage classes; scope of a variable; Formatted I/O; Reading and writing characters; **Operators:** Assignment, Arithmetic, Relational, Logical, Bitwise, Ternary, Address, Indirection, Sizeof, Dot, Arrow, Parentheses operators; Expressions: Operator precedence, Associative rules.

UNIT - III **12Hours**

CONTROL STATEMENTS: Introduction to category of control statements; Conditional branching statements: if, if - else, nested-if, if – else ladder; Switch; Iterative statements: for, while, do - while, Nested loops; Break; Jump; goto and continue.

UNIT - IV **12Hours**

ARRAYS: Introduction; Types of arrays; Single dimensional array: Declaration, Initialization, Usage, Reading, Writing, Accessing, Memory Representation, Operations; Multidimensional arrays.

UNIT - V **12Hours**

FUNCTIONS: User-defined functions; Function declaration: Definition, Header of a function, Body of a function, Function invocation; Call by value; Call by address; Passing arrays to functions; Command line arguments; Recursion; Library Functions.

LABORATORY EXPERIMENTS

Experiment 1: Write a C program to display a simple text on the standard output device using puts ().

Experiment 2: Every character holds an ASCII value (an integer number in the range of 0 to 255) rather than that character itself, which is referred to as ASCII value. Likewise, for a given input whether it is character or digit or special character or lower case or upper-case letter, find corresponding ASCII value.

Experiment 3: Write a C program to swap the two integers with and without using additional variable. **Example**: Before swapping values of a =4, b = 5 and after swapping a = 5, b = 4.

Experiment 4: Write a C program to check whether a given character is a vowel or consonant. **Hint**: Read input from the user, and check whether it is an alphabet or not. If it is an alphabet, then check whether it is a vowel or a consonant. Otherwise display it is not an alphabet.

Experiment 5: The marks obtained by a student in 'n' different subjects are given as an input by the user. Write a program that calculates the average marks of given 'n subjects and display the grade.

Experiment 6: Write a C Program to find the greatest factor of a given input other than itself.
Example: Consider, 30 is the given input, its greatest factor is 15.

Experiment 7: (a) Write a C program to check whether a given number is an Armstrong number or not.
Hint: An Armstrong number is a number which is equal to the sum of digits raised to the power of total number of digits in the number.

Example: Consider the Armstrong numbers are: $0(0^1)$, $1(1^1)$, $2(2^1)$, $3(3^1)$, $153(1^3+5^3+3^3=153)$, $370(3^3+7^3+0^3)$, $407(4^3+0^3+7^3)$, etc.
(b) Write a C Program to print the series of prime numbers in the given range.
Hint: The given number is prime if it is divisible only by one and itself.
Example: if the range is 5 and 15, return 5, 11 and 13 as the series of prime numbers in the given range.

Experiment 8: (a) Write a C Program to print Floyd triangle for the user given number of rows. If the user entered 4 rows, then the output follows:

1
2 3
4 5 6
7 8 9 10

(b) Write a C Program to print the given number of times in a row to form a diamond shape.

Experiment 9: Write a C Program to check whether the given number is a palindrome or not.
Hint: To check whether a number is a palindrome or not, reverse the given number and compare the reversed number with the given number, if both are same then the number is palindrome otherwise not.
Example: Given Number = 121, Reversed number = 121. Hence, given number is palindrome.

Experiment 10: Write a program to search for a given number in the given list of numbers.
Example: Read set of numbers L={2,4,6,1}. Search whether 4 is present in the given list or not.
(b)Write a program to reverse the given list, of size n.
Example: If the list, L= [1,2,3], after reversing it, the list should be, L= [3,2,1]

Experiment 11: Write a C program to perform addition, subtraction, multiplication operations on the two given matrices using functions.

Experiment 12:
Write a C program to swap two numbers using call by value and call by reference.

TEXTBOOKS:

1. Behrouz A. Forouzan, Richard F. Gilberg, "Programming for Problem Solving", 1st edition, Cengage, 2019.
2. Ajay Mittal, "Programming in C - A practical Approach", 1st edition, Pearson Education, India, 2010.

REFERENCE BOOKS:

1. Reema Thareja, "Computer Fundamentals and Programming in C", 1st edition, Oxford University Press India, 2013.
2. Herbert Schildt, "C: The Complete Reference", 4th edition, Tata McGraw-Hill, 2017.
3. Byron S Gottfried, "Programming with C", 4th edition, Tata McGraw-Hill, 2018.