# 8. Basic Programming in C

## (Skill Enhancement Course)

---ByDept of Computer Science, GU

**1. Learning Outcomes:** After completing this course, the students will be

- Familiar with what a programming language is
- Familiar with flowchart and pseudo code
- Familiar with the constructs of C programming languages
- Capable of writing basic C programs

**2. Prerequisites:** NIL

**3. Semester:** 1

**4. Course type:** Skill Enhancement Course

**5. Course level:** 100-199

**6. Theory credit:**2

**7. Practical credit:** 1

**8. Number of required hours:**

**a) Theory:** 30 hrs (30classes)

**b) Practical:** 30 hrs (15 classes)

**9. Reference books:**

- B.S. Gottfried, “Schaum's Outline of Theory and Problems of Programming with C”, Mcgraw-Hill, 2007.
- B. Kernighan, D. Ritchie, “The C Programming Language”, Second Edition, Prentice Hall, 1988
- E. Balaguruswami, “Programming in ANSI C”, 2nd Ed., Tata McGraw Hill, 2004.

**10. Contents of Syllabus:**

**Unit 1: Programming Basics** **(3 Lectures)**

Introduction to programming languages. Low-level and high-level language and their characteristics. Compiler vs. interpreter. IDE. Bugs and its types. Algorithms, pseudocodes and flowcharts. Overview of the C programming language. Structure of a C program.

## Unit 2: Data types and Operators (6 Lectures)

Basic data types in C - integers, floats, doubles, characters, and void. Size and range of values of data types. Variables. Declaring variables. Operators and expressions, Input and output statements – getchar(), getc(), getch(), putchar(), putc(), puts(), scanf(), printf(), format specifiers. Typecasting. Operators in C – binary and unary operators. Arithmetic, assignment, logical, comparison, bitwise and conditional operators. Order of precedence of operators. Associativity of operators. Expressions and statements in C. L-value and R-value. Basic syntax and semantics for expressions and statements.

## Unit 3: Control Structures, Functions and Header files (8 Lectures)

Control structures in C. Decision making with if, if-else, switch statements. Nested conditions. Looping with while, do-while, and for statement. Break and continue statements. Nested loops. Introduction to functions. Function prototypes and arguments. Defining and calling functions in C. Return values and types. Formal and actual parameter. Call by value, Call by reference. Introduction to recursion. Writing recursive functions in C. Importance of main() function, return type of main() function. Preprocessor directives. Include and Define statements. Header files.

## Unit 4: Arrays and Strings (4 Lectures)

Introduction to arrays. Declaration and initialization of arrays. Accessing array elements. Multidimensional arrays. Introduction to strings. Declaration and initialization of strings. String input and output in C. String manipulation functions in C – strlen(), strcpy(), strcat(), strcmp().

## Unit 5: Pointers and Memory Allocation (3 Lectures)

Introduction to Pointers. Pointer declaration and initialization. Pointers and addresses. Pointers and arrays. Pointers and functions. Review of call by reference. Pointer arithmetic.

## Unit 6: Structure and Union (3 Lectures)

Introduction to structures. Declaration and initialization of structures. Accessing structure members. Nested structures and arrays of structures. Unions in C. Declaration and initialization of unions. Accessing union members. Differences between structures and unions.

## Unit 7: File Handling and Preprocessor Directives (3 Lectures)

Introduction to file handling in C. Opening and closing files – fopen(), fclose(). Modes of opening a file. Binary files and text files. Reading and writing files – fgetc(), fgets(), fread(), fputc(), fputs(), fwrite(). File pointers.

## List of Practical

(This is a suggestive list only. Problems need not be restricted to this list.)

1. Write a program in C to print "Hello World"
2. Write a program to take input of two numbers and print their sum, product and difference.
3. Write a program to find the smallest or greatest of three numbers given as input.
4. Write a program to compute simple interest from user given inputs.
5. Write a program to compute factorial of a user given number.
6. Write a program to print the sum and product of digits of an integer.
7. Write a program to print a triangle of stars as follows (take number of lines from user as input):

```
    *
   ***
  *****
 *******
*********
```

8. Write a program to reverse a number.
9. Write a program to compute the sum of the first n terms of the following series
   S = 1+1/2+1/3+1/4+……
10. Write a program to compute the sum of the first n terms of the following series
    S =1-2+3-4+5……
11. Write a function that checks whether a given string is Palindrome or not.
12. Write a function to find whether a given no. is prime or not.
13. Write a program to compute the factors of a given number.
14. Write a program that accepts 10 numbers from the user, stores the numbers in an array and finally displays the maximum and minimum of the numbers.
15. Write a program to perform following operations on strings:
    a) Convert all lowercase characters to uppercase
    b) Convert all uppercase characters to lowercase
    c) Calculate number of vowels in the string
    d) Reverse the string
16. Write a program to implement struct in C. Create a structure of Student with RNo, Name and other credentials with proper datatype and print the same.
17. Write a program to implement union in C. Create a structure of Person with Pid, Name and other credentials with proper datatype and print the same.
18. Write a C program that opens a file for reading and displays the contents of the file in binary mode and text mode.

19. Write a C program that opens a file for reading and displays the contents of the file character by character and line by line on the screen.
20. Write a C program to open a file and count the number of characters and lines in the file.
21. Write a C program that opens a file in append mode and allows the user to add text to the end of the file.

***************************