

Detailed Syllabus :

Course Title: C Programming
Course no: CSC110
Nature of course: Theory + Lab

Full Marks: 60+20+20
Pass Marks: 24+8+8
Credit hours: 3

Course Description: This course covers the concepts of structured programming using C programming language.

Course Objective: This course is designed to familiarize students to the techniques of programming in C.

Course Contents:

Unit 1. Problem Solving with Computer

2 Hrs.

- 1.1 Problem analysis (*requirement analysis, program design, program coding, program testing, software installation and maintenance*)
- 1.2 Algorithms and Flowchart (*symbols start/stop, read/print, processing statement, condition check, direction of flow, connectors*)
- 1.3 Coding, Compilation and Execution (*compiler, integrated development environment, compiling and linking*)
- 1.4 History of C,
- 1.5 Structure of C program (*preprocessor directive, #include and #define directives, header files and library files*)
- 1.6 Debugging, Testing and Documentation (*compiler error, linker error, and run-time error*)

Unit 2. Elements of C

4 Hrs.

- 2.1 C Standards(ANSI C and C99),
- 2.2 C Character Set (*letters, digits, special characters and white spaces*),
- 2.3 C Tokens (*keywords, identifiers, operators, constants, and special symbols*),
- 2.4 Escape sequence,
- 2.5 Delimiters,
- 2.6 Variables,
- 2.7 Data types (Basic, Derived, and User Defined),
- 2.8 Structure of a C program,
- 2.9 Executing a C program,
- 2.10 Constants/ Literals,

2.11 Expressions, Statements and Comments.

Unit 3. Input and Output

2 Hrs.

- 3.1 Conversion specification,
- 3.2 Reading a character,
- 3.3 Writing a character,
- 3.4 I/O operations,
- 3.5 Formatted I/O

Unit 4. Operators and Expression

4 Hrs.

- 4.1 Arithmetic operator,
- 4.2 Relational operator,
- 4.3 Logical or Boolean operator,
- 4.4 Assignment Operator,
- 4.5 Ternary operator,
- 4.6 Bitwise operator,
- 4.7 Increment or Decrement operator,
- 4.8 Conditional operator,
- 4.9 Special Operators(sizeof and comma),
- 4.10 Evaluation of Expression (*implicit and explicit type conversion*),
- 4.11 Operator Precedence and Associativity.

Unit 5. Control Statement

4 Hrs.

- 5.1 Conditional Statements,
- 5.2 Decision Making and Branching (*if, if else, nested if else, else if ladder, and switch statements*)
- 5.3 Decision Making and Looping (*for, while, and do while loops*)
- 5.4 Exit function,
- 5.5 Break and Continue.

Unit 6. Arrays

6 Hrs.

- 6.1 Introduction to Array,
- 6.2 Types of Array (Single Dimensional and Multidimensional),
- 6.3 Declaration and Memory Representation of Array,
- 6.4 Initialization of array,

- 6.5 Character Array and Strings,
- 6.6 Reading and Writing Strings,
- 6.7 Null Character,
- 6.8 String Library Functions(string length, string copy, string concatenation, string compare)

Unit 7. Functions

5 Hrs.

- 7.1 Library Functions,
- 7.2 User defined functions,
- 7.3 Function prototype, Function call, and Function Definition,
- 7.4 Nested and Recursive Function,
- 7.5 Function Arguments and Return Types,
- 7.6 Passing Arrays to Function,
- 7.7 Passing Strings to Function,
- 7.8 Passing Arguments by Value, Passing Arguments by Address,
- 7.9 Scope visibility and lifetime of a variable, Local and Global Variable,

Unit 8. Structure and Union

5 Hrs.

- 8.1 Introduction,
- 8.2 Array of structure,
- 8.3 Passing structure to function,
- 8.4 Passing array of structure to function,
- 8.5 Structure within structure (Nested Structure),
- 8.6 Union,
- 8.7 Pointer to structure

Unit 9. Pointers

6 Hrs.

- 9.1 Introduction,
- 9.2 The & and * operator,
- 9.3 Declaration of pointer,
- 9.4 Chain of Pointers,
- 9.5 Pointer Arithmetic,
- 9.6 Pointers and Arrays,
- 9.7 Pointers and Character Strings,
- 9.8 Array of Pointers,
- 9.9 Pointers as Function Arguments,

- 9.10 Function Returning pointers,
- 9.11 Pointers and Structures,
- 9.12 Dynamic Memory Allocation

Unit 10. File Handling in C

4 Hrs.

- 10.1 Concept of File,
 - 10.2 Opening and closing of File (*naming, opening, and closing a file*)
 - 10.3 Input Output Operations in File (*reading data from file, writing data to a file*)
 - 10.4 Random access in File (*ftell(), fseek(), rewind()*)
 - 10.5 Error Handling in Files (*feof(), ferror()*)
- (Note: *address some of the functions associated with file handling, e.g. fopen(), fclose(), fgetc(), fputc(), fprintf(), fscanf()*)

Unit 11. Introduction to Graphics

3 Hrs.

- 11.1 Concepts of Graphics (*graphics.h header file*)
- 11.2 Graphics Initialization and Modes (*graphics driver and graphics mode*)
- 11.3 Graphics Function (*Basic functions of graphics.h e.g. line(), arc(), circle(), ellipse(), floodfill(), getmaxx(), getmaxy()*)

Text Books:

1. Byron Gottfried: "Programming with C," , Second Edition, McGraw Hill Education.
2. Herbert Schildt, C The Complete Reference, Fourth Edition, Osborne/McGraw-Hill Publication.

Reference Books:

1. Paul Deitel, Harvey Deitel, C: How to Program, Eighth Edition, Pearson Publication.
2. Al Kelley, Ira Pohl: "A Book on C", Fourth Edition, Pearson Education.
3. Brian W. Keringhan, Dennis M. Ritchie, The C programming Language, Second Edition, PHI Publication.
4. Ajay Mittal, Programming in C: A Practical Approach, Pearson Publication
5. Stephen G. Kochan, Programming in C, CBS publishers & distributors.
6. E. Balagurusamy, Programming in ANSI C, Third Edition, TMH publishing

Details of Laboratory work in C programming

Laboratory Works: This is the first “programming” course in B.Sc.CSIT. It builds the foundation on how to write a program using any high level language. Hence, this course requires a lot of programming practice so that students will be able to develop good logic building and program developing capability which is essential throughout the B.Sc.CSIT course and thereafter. 20% of the total marks is assigned from the practical. Some important contents that should be included in lab exercises are as follows:

Unit 1:

2 Hrs.

Create, compile, debug, run and test simple C programs

Unit 2,3,4:

5 Hrs.

Using different data types available in C, perform arithmetic operations in C, perform formatted input/output operations, perform character input/output operations.

Using relational operator, logical operator, assignment operator, ternary operator, and other operators. Evaluation of Expression to check operator precedence and associativity.

Unit 5:

6 Hrs.

Create decision making programs using control statements like; if, if..else, if..else ladder, nested if, and switch cases.

Create programs using loops (for, while, do while, nested loops) and realize the differences between entry controlled and exit controlled loops.

Unit 6:

6 Hrs.

Create, manipulate arrays and matrices (single and multi-dimensional), work with pointers, dynamically allocate/de-allocate storage space during runtime, manipulate strings (character arrays) using various string handling functions.

Unit 7:

6 Hrs.

Create user-defined functions with/without parameters or return type, create recursive functions, use function call by value and call by address, work with automatic, global and static variables.

Unit 8:**5 Hrs.**

Create and use simple structures, array of structures, nested structure. Passing structure and array of structure to function, concept of pointer to structure

Unit 9:**6 Hrs.**

Create programs that addresses pointer arithmetic, pointers and arrays, pointer and character strings, pointers and functions, pointer and structure, and dynamic memory allocation.

Unit 10:**5 Hrs.**

Create files that address random access and input/output operations in file, create files to keep records and manipulation of records etc.

Unit 11:**4 Hrs.**

Create graphics program that address some basic functions of graphics.h header file, e.g. line(), arc(), circle(), ellipse(), floodfill(), getmaxx(), getmaxy() etc.

Note: Motivate students to create small project work integrating all of the above concepts.

Model Question**Section A****Long Answer Questions**

Attempt any 2 questions. [2*10=20]

1. Define function and list its advantages. Describe the difference between passing arguments by value and passing arguments by address with suitable program. [4+6]
2. Explain how structure is different from union? Make a program using structure of booklist having data member's title, author, and cost. Enter four data and calculate total cost. [3+4+3]
3. Explain various modes in which file can be opened? Write a program to CREATE and WRITE N numbers in a file "NUMBER.TXT". Open this file then read its content and put all even numbers in one file "EVEN.TXT" and odd numbers in another file "ODD.TXT". [2+4+4]

Section B**Short Answer Questions**

Attempt any 8 questions. [8*5=40]

4. What do you mean by a problem analysis? What are the properties of a good algorithm? Explain the Compilation and Execution of any C program? [1+1+3]

5. Define nested if else statement with suitable flowchart. Write a C code to check if user given input is exactly divisible by 5 or 11 using nested if else statement? [2+3]
6. List various binary and unary operators used in C? Write a program that uses a “while” loop to compute and prints the sum of a given numbers of squares. For example, if 4 is input, then the program will print 30, which is equal to $1^2+2^2+3^2+4^2$. [1+4]
7. “Size of character array is always declared one more than the input size.” Justify the statement. Write a program to read a character array input as “TRIBHUVAN UNIVERSITY” from the user and find out how many times a character ‘I’ occurs in that array? [1+4]
8. Write syntax to declare and initialize 2-dimensional array? With suitable program logic explain how would you find transpose of a 3*3 matrix? [1+4]
9. Explain the concept of recursive function using the example program to find the factorial of given positive integer. [5]
10. Describe the fundamental concept of pointer and its arithmetic with suitable examples. [5]
11. Explain the use of graphical functions. Write a program to draw a triangle using line() graphics function. [1+4]
12. Write short notes on: [2+3]
 - i) Dynamic Memory Allocation
 - ii) break and continue