COMP 102: COMPUTER PROGRAMMING

(Course plan)

Instructor: Sunil Regmi

Prerequisite: Students should have fundamental knowledge about computer.

All the programming assignments and labs we do during this semester are in C programming language.

Objectives: In this course you will learn how to program basically using C programming language. It also encourages the students to write simple to moderate program. This course also provide base for Object Oriented programming.

Grading Policy:

Internal: 50

Internal-I	10
Internal-II	10
MCQ’s	5
Lab Reports	5
Assignments	5
Final Lab Exam (Practical) || PPT	10
Viva	5
Total	**50**

Final: 50

- **10 marks MCQ**
- **40 marks subjective**

80% attendance compulsory in order to be eligible to appear in FINAL EXAM

Total Lecture Hours: 15 Weeks X 4 Hours = 60 Hours

Classes:

- **Sunday and Monday – 1:00pm to 3:00 pm**

Text Books:

1. E Balaguruswamy (2016). Programming in ANSI C, McGraw Hill Education, India.
2. Daya Sagar Baral, Diwakar Baral and Sharad Kumar Ghimire &quot;The Secrets of C Programming Language&quot;, Bhundipuran Publication
3. A L Kely &amp; Ira Pohl “A Book on C” Benjamin/Cumming
4. Brian W. Keringhan &amp; Dennis M. Ritchie (1988) “The C Programming Language”. Prentice Hall

SYLLABUS

COMP 102 [STRUCTURED PROGRAMMING]

Credit Hours: - 3

1. Introduction (2 hrs)

1.1 History of C
1.2 Introduction to C
1.3 Types of Programming Languages,
1.4 Types of error
1.5 Basic Structure of C program

2. Problem Solving Using Computer (2 hrs)

2.1 System development life cycle (Problem analysis, Algorithm development and Flowchart, Compilation and Execution, Debugging and Testing, Programming Documentation)

3. Element of C (3 hrs)

3.1 The C Character set
3.2 Identifier and Keywords
3.3 Data Type, Variable Declaration
3.4 Constants (String, Numeric, Character Constant)
3.5 Symbolic Constant

4. C Operators (4 hrs)

4.1 Arithmetic Operators
4.2 Assignment Operators
4.3 Logical and Comparison Operations
4.4 Bitwise Operators
4.5 Unary Operator
4.6 Conditional Operators

5. Writing a Program in C (Data Input and Output) (2 Hours)

5.1 Simple Program
5.2 Input Statement
5.3 Output Statement
5.4 Features of stdio.h

6. Control Structure (7 hrs)

6.1 Condition Statements
- a. if Statement
- b. if-else Statement

        c. switch statement

    6.2 Loop Statements

        a. for loop
        b. while loop
        c. do-while loop

    6.3 Break Control Statements

        a. break
        b. continue
        c. go-to statement

7. Functions (6 hrs)

7.1 Defining Function
7.2 Use of function
7.3 Function Prototypes
7.4 Passing Argument to a Function
7.5 Recursive function
7.6 Storage Class

8. Arrays and Strings (6 hrs)

8.1 Defining an Array
8.2 Processing an Array
8.3 Passing Array to Function
8.4 Multidimensional Array

9. Pointers (6 hrs)

9.1 Pointer Declaration
9.2 Pointer Arithmetic
9.3 Operation on Pointers
9.4 Pointer and Array (Pointer and one dimension Array)
9.5 Dynamic Memory Allocation

10. Structure (6 hrs)

10.1 Declaration of Structure
10.2 Initialization of Structure
10.3 Array of Structure
10.3 Pointer to Structure

11. File Handling (4 hrs)

11.1 Introduction
11.2 Defining and Opening a File (create, open, read, write, move, close)

LESSON PLAN

Week	Topics	Assignments	Labs
1	Getting Started		
	Introduction of C		
2	Problem Solving Using Computer Elements of C		
3	Elements of C contd.. C operators	1	1
4	C operators contd.. Writing a Program in C (Data Input and Output)		
5	Control Structure		2
6	Control Structure contd..		3
7	Control Structure contd.. Arrays and Strings	2	4
8	**First Internal Assessment**		
9	Arrays and Strings contd..	3	5
10	Functions		6
11	Functions contd.. Pointers		7
12	Pointers contd..	4	8
13	Structure		9
14	Structure contd.. File Handling	5	10
15	**Second Internal Assessment**		

ASSIGNMENTS

Assignment #1:

1. Define programming language. Give a short overview of Machine Language, Assembly Language and High-Level Language.
2. List the characteristics of C Programming Language. Who developed the C programming language, when and where?
3. What are the advantages of High-Level Programming Language?
4. Why do we need flowchart, algorithm and pseudocode in computer programming? Explain with examples.
5. What is a compiler, why do we need it for computer programming? List common compilers for C programming.
6. What are errors in a computer program, how do you overcome them?
7. Explain the structure of C programming language.
8. Explain the steps of writing and executing a C Program.
9. List out the desirable characteristics of computer programs.

Assignment #2:

1. What are Character Sets? List different types of character sets supported by C programming language.
2. Differentiate between identifiers and keywords.
3. What are the rules for writing variable names in C?
4. Why is it important to select correct data types in programming? List the types of data types commonly used in C programming language.
5. Typically, what is the largest permissible magnitude of floating-point constant? Compare with an integer constant.
6. What is a constant? Explain Sting, Numeric, Character Constant with suitable examples.
7. When writing integer constants, how are decimal constants, octal constants and hexadecimal constants distinguished from one another?
8. What is an escape sequence? What is its purpose, give examples?
9. What are unsigned integer constants? What are long integer constants? How do these constants differ from ordinary integer constants? How can they be written and identified?
10. What happens to the symbolic constants that appear with in a C program during compilation process.

Assignment #3:

1. Define operators. List out the operators used in C Programming.
2. Describe the five arithmetic operators in C. Summarize the rules associated with their use.
3. Present a brief account of increment and decrement operators with examples.
4. Define operator precedence. What are the relative precedence of the relational, quality and logical operators with respect to one another and with respect to arithmetic and unary operators? What is their associativity’s?

5. What are library functions? Why are they important?

6. Suppose x, y and z are floating-point variables that have been assigned the values x= 8.8, y=3.5 and z= -5.2. Determine the values of each of the following arithmetic expressions:

a) x+y+z
b) x/(y+z)
c) 2*y+3*(x-z)
d) (x/y) +z
e) x % y
f) 2*x/(3*y)

7. List out the commonly used input/output functions in C. What is the purpose of scanf function? How is it used in a C Program? Compare scanf with getchar function.

Assignment #4:

1. Compare the use of the if-else statement with the use of conditional operator.
2. What is the purpose of the do-while statement? How does it differ from the while statement?
3. State at least three advantages of making your program modular using functions.
4. What is Recursion? What are the advantages of using recursion?
5. What three types of errors do function-prototypes help prevent?
6. What is meant by the scope of a variable within a program?

Assignment #5:

1. Differentiate between pass by value and pass by reference with an example.
2. What is a pointer variable? What are the advantages of using a pointer?
3. What is the difference between array and pointer variable? In what way are they similar?
4. What is a structure? How is it different from union?
5. What is DMA? Why are they important? How can we apply DMA?

LAB WORKS

Lab #1

1. Write a C program to display “This is my first C Program”.
2. Write a C program to add two numbers (2 and 6) and display its sum.
3. Write a C program to multiply two numbers (4 and 5) and display its product.
4. Write a C program to calculate area and circumference of a circle.
5. Write a C program to perform addition, subtraction, division and multiplication of two numbers.

Lab #2:

1. Write a program to calculate simple and compound interest.
2. Write a program to swap values of two variables with and without using third variable.
3. Write a program to display the size of every data type using “sizeof” operator.
4. Write a program to illustrate the use of unary prefix and postfix increment and decrement operators.
5. Write a program to input two numbers and display the maximum number.
6. Write a program to find the largest of three numbers using ternary operators.

Lab #3:

1. Write a program to read a sentence and counts the total number of character (excluding space) using while loop.
2. Write a program to generate Fibonacci sequence using do while loop.
3. Write a program to read number and identifies whether the given number is a prime number or not.
4. Write a program to check whether a given number is Armstrong or not. 371 is Armstrong because 3 * 3 * 3 + 7 * 7 * 7 + 1 * 1 * 1 = 371, Other Armstrong numbers to check are 153 and 370

Lab #4:

1. Write a Program to Check Whether a Number is Prime or not.
2. Write a program to find the largest and smallest among three entered numbers and also display whether the identified largest/smallest number is even or odd.
3. Write a program to compute grade of students using if else adder. The grades are assigned as followed:

Marks	Grade
a. marks<50	F
b. 50≤marks< 60	C
c. 60≤marks<70	B
d. 70≤marks<80	B+
e. 80≤marks<90	A
f. 90≤mars≤ 100	A+

4. Write a program to check whether the entered year is leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400.)
5. Write a program to find the factorial of a number.
6. Write a program to check number is Armstrong or not.
(Hint: A number is Armstrong if the sum of cubes of individual digits of a number is equal to the number itself).

Lab #5

1. Write a program to count number of digits in a given integer.
2. Write a program to reverse a given integer.
3. Write a program to print number in reverse order with a difference of 2.
4. Write a program to print the sum of digits of a number using for loop.
5. Write a program to check whether a number is Palindrome or not.
6. Write a program to generate Fibonacci series

Lab #6

1. Write a Program to Search an element in array.
2. Write a Program to perform addition of all elements in Array.
3. Write a Program to find the largest and smallest element in Array.
4. Write a Program to reverse the array elements in C Programming.
5. Write a Program for deletion of an element from the specified location from Array.
6. Write a Program to access an element in 2-D Array.
7. Write a program for addition of two matrices of any order in C.

Lab #7

1. Write a program to add, subtract, multiply and divide two integers using user defined type function with return type.
2. Write a program to calculate sum of first 20 natural numbers using recursive function.
3. Write a program to generate Fibonacci series using recursive function.
4. Write a program to swap two integers using call by value and call by reference methods of passing arguments to a function.
5. Write a program to find sum of digits of the number using Recursive Function.
6. Write a program to read an integer number and print the reverse of that number using recursion.
7. Write a C program to find maximum and minimum between two numbers using functions.

Lab #8

1. Write a program to find the sum of all the elements of an array using pointers.
2. Write a program to swap value of two variables using pointer.
3. Write a program to add two numbers using pointers.
4. Write a program to input and print array elements using pointer.
5. Write a program to copy one array to another using pointer.

6. Write a program to swap two arrays using pointers.
7. Write a program to reverse an array using pointers.

Lab #9

1. Write a program to create a structure named company which has name, address, phone and noOfEmployee as member variables. Read name of company, its address, phone and noOfEmployee. Finally display these members‟ value.
2. Define a structure “complex” (typedef) to read two complex numbers and perform addition, subtraction of these two complex numbers and display the result.
3. Write a program to read RollNo, Name, Address, Age & average-marks of 12 students in the BCT class and display the details from function.
4. Write a program to add two distances in feet and inches using structure.
5. Write a program to read and print an Employee‟s Details using Structure.

Lab #10

1. C Program to list all files and sub-directories in a directory.
2. C Program to count number of lines in a file.
3. C Program to print contents of file.
4. C Program to copy contents of one file to another file.
5. C Program to merge contents of two files into a third file.
6. C program to delete a file.