

COMP 102: COMPUTER PROGRAMMING

Year: I

Semester: II

Credit: 3

Course Description:

This course introduces the fundamental concepts of procedural programming in C. Topics include data types, control structures, functions, arrays, structure and pointer. This course also focuses on the development of problem solving skills using programs.

Prerequisite:

Students should have fundamental knowledge about computer. All the programming assignments and labs during the semester are in C programming language.

CHAPTERS:

- | | |
|--|----------------------|
| 1. Introduction | [1 Hour] |
| 1.1. History of C | |
| 1.2. Introduction to C | |
| 1.3. Important of Program (C Program) | |
| 1.4. Desired Program Characteristics | |
|
2. C Fundamentals |
[3 Hours] |
| 2.1. The C Character set | |
| 2.2. Identifier and Keywords | |
| 2.3. Data Type, Variables, Declaration | |
| 2.4. Constants (String, Numeric, Character Constant) | |
| 2.5. Symbolic Constant | |
|
3. C Operators |
[4 Hours] |
| 3.1 Arithmetic Operators | |
| 3.2 Assignment Operators | |
| 3.3 Logical and Comparison Operations | |
| 3.4 Bitwise Operators | |
| 3.5 Unary Operator | |
| 3.6 Conditional Operators | |
|
4. Writing a Program in C (Data Input and Output) |
[4 Hours] |
| 4.1 Simple Program | |
| 4.2 Input Statement | |
| 4.3 Output Statement | |
| 4.4 Features of stdio.h | |

5 Control Structure [8 Hours]

5.1 Condition Statements

- a. if Statement
- b. if-else Statement
- c. switch statement

5.2 Loop Statements

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

5.3 Break Control Statements

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

6 Function [5 Hours]

6.1 Defining Function

6.2 Use of function

6.3 Function Prototypes

6.4 Passing Argument to a Function

6.5 Recursive function

7 The Scope of Variable [4 Hours]

7.1 Storage Class

7.2 Automatic Variable

7.3 Static Variable

7.4 External Variable

8 Array [6 Hours]

8.1 Defining an Array

8.2 Processing an Array

8.3 Passing Array to Function

8.4 Multidimensional Array

9 Pointer [6 Hours]

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 **Program Structure**

[4 Hours]

- 10.1 Declaration of Structure
- 10.2 Initialization of Structure
- 10.3 Array of Structure

Books:

Programming with C
Byron Gottfried

Programming with C
K.R. Venugopal
Surdeep R. Prasad

COMP 102: Computer Programming

Assignments

Assignment #1: Operators, Expression

Question 1: What is meant by operator precedence? Illustrate with an example.

Question 2: What is meant by associativity? Illustrate with an example.

Question 3: What are library functions? Why are they important? List out ten library functions that are frequently used in C program. What are header files in C?

Question 4: What are keywords? List out 15 keywords that are frequently used in C program.

Assignment #2: Branching, looping

Question 1: Compare the use of the if-else statement with the use of conditional operator.

Question 2: What is the purpose of the default keyword used for switch case? Is it possible to compare else if ladder with switch case?

Question 3: What is the purpose of do-while statement? How does it differ from the while statement?

Question 4: What is the purpose of the break and continue statement?

Assignment #3: Functions

Question 1: State at least three advantages of making your program modular using functions.

Question 2: What is Recursion? What advantages is there in its use? What are its disadvantages?

Question 3: Compare library function and user defined functions. What is the purpose of user defined function?

Question 4: What three types of errors do function-prototypes help prevent?

Question 5: What is meant by the scope of a variable within a program?

Assignment #4: Array

Question 1: In what way does an array differ from an ordinary variable?

Question 2: Describe element storage mechanism in a 2-D array.

Question 3: How can you pass array in a function? What are the things that are needed to be considered while passing an array to a function?

Assignment #5: Structures, Pointers

Question 1: Differentiate between pass by value and pass by reference with an example.

Question 2: What is a pointer variable? What are the advantages of using a pointer?

Question 3: What is the difference between array and pointer variable? In what way are they similar?

Question 4: What is a structure? How is it different from union?

Question 5: What is DMA? Why are they important? How can we apply DMA?

COMP 102: Computer Programming

Lab Exercises

Week #1

Introducing Turbo C++ Compiler and its environment / Linux Environment

Week #2

Some sample C program.

Some problems related to formatted strings.

Week #3: Operators and expressions

1. Write a program to convert centigrade to Fahrenheit. $[F = 9/5 * C + 32]$
2. Write a program that calculates the area of a circle and circumference.
3. Write a program that calculates the area of a triangle. $\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$ and $s = (a + b + c) / 2$
4. Write a program that reads the marks in each subject and calculates the percentage. Assume 3 subjects have to be read from user.

Week #4: Conditional Statements

1. Write a program that reads a number and identifies whether the given number is even or odd.
2. Write a program to find the largest number among three numbers.
3. Write a program to read the marks of 5 subjects and print the equivalent grade based upon the condition shown below.

Range Grade

≥ 80	and	≤ 100	A
≥ 75	and	< 80	A-
≥ 70	and	< 75	B+
≥ 65	and	< 70	B
≥ 60	and	< 64	B-
≥ 55	and	< 60	C+
≥ 50	and	< 55	C
≥ 45	and	< 50	C-
≥ 40	and	< 45	D

otherwise FAIL

Week #5: Loop

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.

Try this at HOME

1. Write a program to identify whether the given number is a perfect number or not. 28 is a perfect number because $28/2 = 14$ and the sum of the numbers that perfectly divides 28 from 1 to 14 is 28.

2. Write a program to calculate the factorial of a given number.

3. Write a program to check whether a given number is dudeney number or not. 512 is dudeney because $(5+1+2)^3 = 512$. 4913 is another dudeney number.

Week #6: Function

1. Write a program to identify whether the given number is a perfect number or not using a function. 28 is a perfect number.

2. Write a program to evaluate GCD of two given integers. Use function that returns GCD.

3. WAP to reverse a given number. [Reverse of 155 is 551]

Week #7: Function

1. Write a program to find all the prime numbers between 1 to 100.

2. Write a program to check whether a number is strong or not. 145 is a strong number because $1! + 4! + 5! = 145$

Week #8: Recursion

1. Write a recursive program to find the factorial of a given number.

2. Write a recursive program to find the sum of n natural numbers.

3. Write a recursive program to find a GCD of two numbers.

Week #9: Array

1. Write a C program to store N numbers in a one dimensional array and calculate its average with the help of the function.

2. Write a C program to convert a binary number to decimal with the help of the function. `int todecimal(char bits[20], int length)`, here bits is the character array to represent bits of binary numbers and length is the number of bits in the binary number.

Week #10: One Dimensional array and function

1. Write a program to arrange the numbers (array) in ascending order using bubble sort.

2. Write a program to find decimal equivalent of a binary number (store in an array).

Week#11: Multi-dimension array and function

1. Write a program to evaluate transpose of n by n matrix with the help of function.

transpose(int matrix[][20], int n). Here matrix is the matrix to be transformed and n is the dimension of matrix. The function should return transpose of the matrix.

2. Write a C program for matrix addition with the help of function int add(int a[][20], int b[][20], int c[][20], int row, int col)

Here a and b are matrix to be added, c is the resultant matrix after addition and row and col are dimension of a and b. the function should return m by n matrix containing the addition data.

Week #12: Pointers

1. Write a program that swaps two variables. Use function and pointers.
2. WAP to store 10 floating point values in an array and sort them in ascending order. Use function and pointer

Week #13: Structure

1. Write a program that defines a structure called STUDENT with suitable attributes and reads the data for 5 students. Your program should display the records in ascending order according to the roll no of the students.
2. Consider a plane graph. Write a program that uses function to return a distance between given point and the origin.