National University

Of Computer & Emerging Sciences Faisalabad-Chiniot Campus

CL-217
Object Oriented Programming

Lab No 5

Objectives:

- Recursion
- Structures

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential
5. Write a C++ statement(s) for each of the following task one after the other, in the same order.
6. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task outputs in MS word and do not submit .cpp file with word file**.
7. First think about statement problems and then write/draw your logic on copy.
8. After copy pencil work, code the problem statement on MS Studio C++ compiler.
9. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Microsoft teams. (Make sure your submission is completed).
10. Please submit your file in this format 19F1234_L4.
11. Do not submit your assignment after deadline.
12. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem 1: | (Recursion)

Write a recursive function that receives an integer consisting of any number of digits. Your function should calculate and return the summation and average of the integer digits.

Enter number: 5

Total sum: 15

Note: Use minimum lines of code as possible to get max marks.

Problem 2: | (Recursion)

Write a recessive function to print a Fibonacci series up to N numbers

Enter the number of elements: 10

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34

Note: Use minimum lines of code as possible to get max marks.

Problem 3: | (Recursion)

Write a C++recursive function to check out if any given integer is prime or not.

Note: Use minimum lines of code as possible to get max marks.

Problem 4: | (Recursion)

Write a program to perform bubble sort on an array.

1. Size and elements of array should be entered by user.
2. Use recursive function to perform bubble sort

Note: Use minimum lines of code as possible to get max marks.

Problem 5: | (Structure, Structure variable, static memory allocation)

The structure Car is declared as follows:

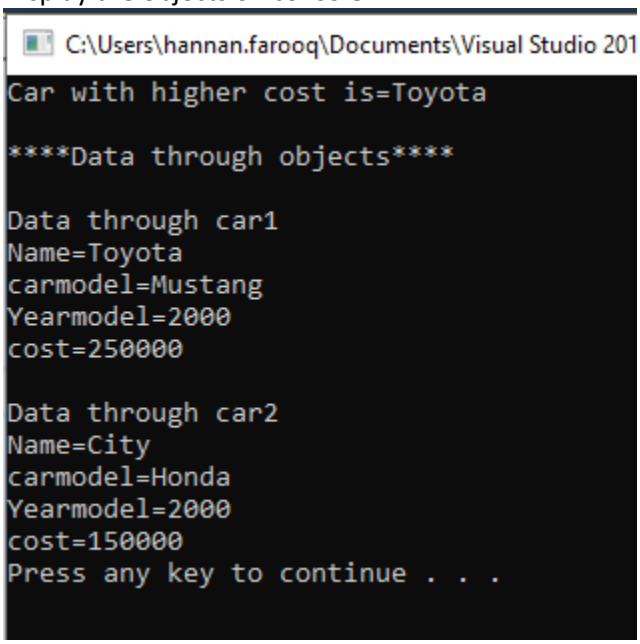
```
struct Car
{
char carName[20];
 char carModel[20];
 int yearModel;
 double cost;
};
```

1. Write a definition statement that defines a Car structure variable car1 and initialized with the following data

Name	Toyota
Model	Mustang

Year Model	2000
Price	$25000

2. Define another object car2 and initialize it with any valid data.
3. Compare the cost of both objects and print the name of car with higher cost.
4. Display the objects on console.



Problem 6: (Structure, Structure object)

Write a program in C++ that shows the area of 3 room's. Using Structure namely "distance".

1. Take input of feet & inches from user for variable d1 (feet & inches),
2. Assign variable d2 = {10, 5.25} values.
3. Now add feet and inches of d1 & d2 and store in d3.
4. Display d1 (feet & inches) d2 (feet & inches) d3 (feet & inches) separately.
5. Put Condition if d1 & d2 inches increase by 12 it become a foot.
6. Display the objects d1, d2 and d3 on console.

Problem 7: (Structure, pointer to structure, Structure variable array, dynamic memory allocation)

Write a program that simulates a soft drink machine. The program should use a structure that stores the following data:

Drink Name

Drink Cost

Number of Drinks in Machine

The program should create a dynamic array of four structures. The elements should be initialized with the following data:

Drink Name	Cost	Number in Machine
Cola	.75	20
Root Beer	.75	20
Grape Soda	.80	20
Cream Soda	.80	20

Each time the program runs, it should enter a loop that performs the following steps:

- A list of drinks is displayed on the screen
- The user should be allowed to either quit the program or pick a drink.
- If the user selects a drink, he or she will next enter the amount of money according to the cost shown into the drink machine.
- The program should display the amount of change that would be returned and subtract one from the number of that drink left in the machine.
- If the user selects a drink that has sold out, a message should be displayed. The loop then repeats.
- When the user chooses to quit the program, it should display the total amount of money the machine earned.

Input Validation: When the user enters an amount of money, do not accept negative values, or values greater than $1.00.

Proper code indentation will hold extra marks !

Best of luck ☺

You are done with your exercise, submit on Teams at given time.