National University of Computer and Emerging Sciences, Lahore Campus

Course:	**OOP Lab**	**Course Code:**	**CL1004**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2021**
Duration:	**2 Hours**	**Total Marks:**	**100**
Paper Date:	**20-Nov-2021**	**Weight:**	
Section:	**H**	**Page(s):**	
Exam:	**Lab Midterm**	**Reg. No.**	

*The word impossible itself says **I'm Possible**.*

Read below Instructions Carefully:

- Understanding the question statement is also part of the exam, so do not ask for any clarification. In case of any ambiguity, make suitable assumptions.
- You have to complete exam in 2 hrs. No extra time will be given for submission.
- For Q1, Submit a single .cpp file named as 19L-9085-Q1.cpp
- For Q2, submit a single file (containing function and main) named as 19L-9085-Q2.cpp.
- Put both file in a folder, rename that folder with your roll number and paste/drop the folder at the mentioned: **\\Cactus\Xeon\Fall 2021\Dilawar Shabbir\MID Exam\Submission\Section H**
- Submit both questions .cpp on Google Classroom under assignment titled as **OOP- Lab Midterm Submission**.
- Your code should be intended and commented properly. Use meaningful variable names.
- It is your responsibility to save your code from being copied. All matching codes will be considered cheating cases. PLAGIARISM will result in forwarding of case to Disciplinary Committee and negative marks in Midterm.

Question No. 01:

Consider the following class that holds the student data.

```
class Student
{
   char *rollNo; //roll no of student like 20L-2383
   int semester; //semester number
   char *name; //name of student
   float CGPA;

// Your class Student should have all the necessary function required to complete the below
task.

// You may add additional data members if required.


 };

int main()
 {
        Student S1,
        S1.Input(); // Takes input from user.
        S1.Output();  /*This function displays the Rollno,name,Cgpa and semester of the
        calling Student object*/

        Student list[5];
        For(int I = 0; i<5 ; i++)
        List[i].Input();

        For(int I = 0; i<5 ; i++)
        List[i].Output();

        SortByCGPA(List);//Global function (Sorts the students on GPA base)

        For(int I = 0; i<5 ; i++)
        List[i].Output();

         return 0;
}
```

Note: You cannot change the above main function.

Sample Output(working) on Console:

// Sample run is given for 2 students data.

```
Enter details of the students.
Enter Roll Number of the Student        : 21L-1122
Enter Current Semester of the Student  : 3
Enter Name of the Student               : Johnson Johnson
Enter CGPA of the Student               : 3.75

Enter details of the students.
Enter Roll Number of the Student        : 21L-1133
Enter Current Semester of the Student  : 3
Enter Name of the Student               : Paul Walker
Enter CGPA of the Student               : 3.57
```

// Once the data has been entered, your output function shows the details of the students.

```
Roll Number  : 21L-1122
Semester     : 3
Name         : Johnson Johnson
CGPA         : 3.75

Roll Number  : 21L-1133
Semester     : 3
```

```
Name        : Paul Walker
CGPA        : 3.57
```

// Now after calling of SortByCGPA function your data of students will be sorted in ascending order and will be shown as the student with least CGPA first and increasing down the list.

```
Roll Number : 21L-1133
Semester    : 3
Name        : Paul Walker
CGPA        : 3.57

Roll Number : 21L-1122
Semester    : 3
Name        : Johnson Johnson
CGPA        : 3.75
```

Question No. 02:

Factor calculator is a program which asks user how many inputs he wants to take and then asks user to enter **n** numbers such that, Numbers entered should be **greater than 6 and less than 100**. If the number does not fall in this range, program should alert user to re-enter the number. If the entered number is -1, the program should stop taking input and return the factors of all the entered numbers. Create factors array for each number in above allocated 2D array and store that number on 1st index of factors array and the factors on the further indexes. Your program should also **de-allocate** acquired dynamic memory at the end of the life cycle. **Do not consume** extra memory.

Int** GetFactorsArray(…) // This function takes count of elements and values of elements from user. Returns a dynamic 2-D Array of Factors of the elements(numbers) entered.

Void Print(…….); Displays 2-D array in proper format.

```
Void main()
{

// Design main as per your requirements.

}
```

Sample Output:

```
Enter no of inputs you want to take: 2
Enter Number between 6 and 100: 12

Enter Number between 6 and 100: 5

Wrong input. Please re-enter the number

Enter Number between 6 and 100: 10
...
...

Following are the factors of numbers entered:


Factors of 12 are 1,2,3,4,6,12
Factors of 10 are 1, 2,5,10
```