Name :		Roll No:	Serial No:	Section :	

CL217-Object Oriented Programming Lab

Monday, June 23, 2020

Final Term Exam
Spring Semester 2020
Max Time: 3 Hour
Max Marks: 40

Course Instructor

Muhammad Haris, Saqib Hayat, M Ali, Mughees Ismail

General Instructions:

1. There are Four (4) Questions in total.
2. Questions reading and understanding are also part of the exam, only answer what is asked.
3. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required.

Guidelines for Submission:

1. **You must solve/attempt it on compiler insert it in provided docx file with screenshots. The code can be pasted inside that document under the appropriate question.**
2. You should submit only one DOCX file [DO not convert to PDF/rar].
3. Naming Convention of file must be followed.
4. Example:
   a. Rollnumber_Section.pdf
   b. 19f0123_C.docx
5. Submissions submitted after the due time shall not be considered.
6. If you don't finish every part of a question, don't worry! You can still submit what you've done to get marks based on your efforts.
7. **In case of copied or plagiarized solutions in exam Or If a student provided help to another student during exam both will be awarded "F" grade and it will affect the student CGPA.**
8. Viva of any student can be conducted by the instructor after conducting an online exam in case of any doubt.
9. You can submit your code on classroom **But in the worst case, you can email it to your class teacher within the deadline.**

	Q-1	Q-2	Q-3	Q-4	Total
Total Marks	**10**	**10**	**10**	**10**	**40**
Marks Obtained					

Question 1:

Define a class **Matrix** which will have the following data members

- Int **arr;
- Int rows;
- Int columns;

Define **default constructor** which will assign the rows=columns=3 and allocate memory to **arr and initialize the contents to 0. It will also print "I am default constructor".

Define **overloaded constructor** which will take rows and columns as a parameter and allocate memory to **arr and initialize the contents to 0. It will also print "I am overloaded constructor with rows = *"passed value"* and column = *"passed value"* ".

Define destructor

Overload following operators

- **>>** to input values for matrix object using friend function
- **<<** to output values of matrix object using friend function

Now you need to define a unary operator **!** to check if the matrix is Identity or not.'

In main do the following steps:

```
Int main(){
        Matrix A;
        Matrix B(4,4);
        cin>> A;
        cout<<A;
        If(!A)
        {
        cout<<"it is Identity matrix" <<endl;
        }
        else
        {
        cout<<"it is not Identity matrix"<<endl;
        }
}
```

Identity matrix

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Paste Code and Screenshot here

Question 2:

You are required to make a C++ program that will calculate the area of **shape** after taking input from the user. There are a few shapes that your program will support:

1. Circle (A = πr2)
2. Rectangle (A = wl)
3. Triangle (A = 1/2*(b*h))
4. Pentagon

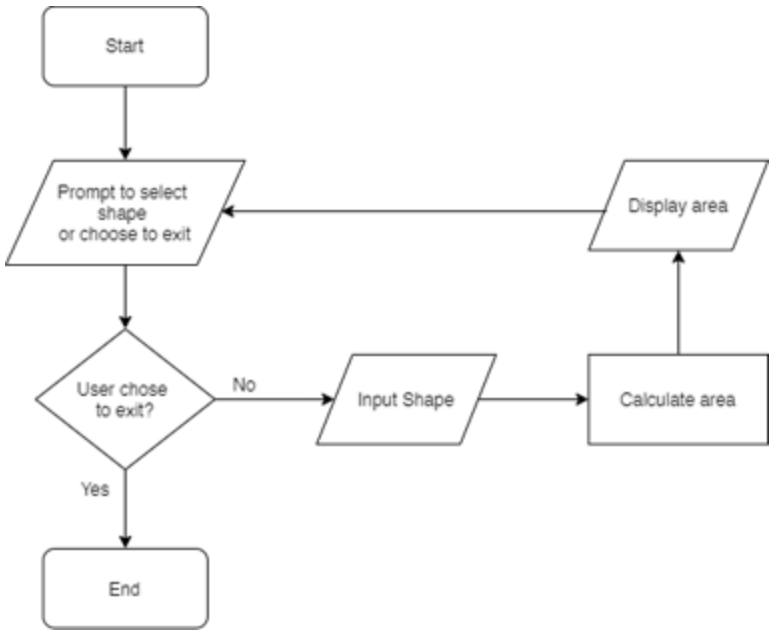
Your main function will prompt the user to choose the type of shape and then ask the user for the required parameters to calculate its area. Once the program successfully calculates the area of the shape it will be displayed on the screen and then the program will ask the user if they want to calculate the area of another shape otherwise, the program will end if the user chooses not to.

Your program must be extensible; i.e. more shapes may be added later on to your software. Use the principles of OOP to design your solution.

You can only create one instance of a class in your whole program, creating more than one instance will get you zero marks.

Note: The major marks of this question are the design and correct use of the concepts of Object-Oriented Programming. Recall the concepts taught to you during the semester and use them where necessary.

The general flow of the program would be like this:



-

Paste Code and Screenshot here

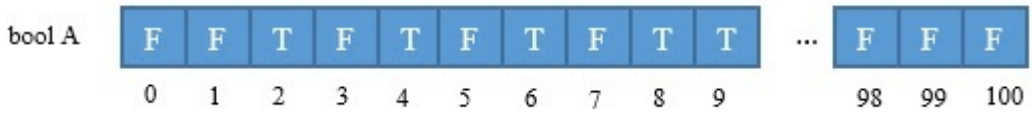
Question 3:

Implement via 3 file structure.

(IntegerSet Class)

- Create class IntegerSet for which each object can hold any number of integers in the range 0 through 100.
- Represent the set internally as an array of bool values [not actual inputs]. Element a[i] is true if integer i is in the set. Element a[j] is false if integer j is not in the set.

$$A = \{2,4,6,8,9 \ldots\}$$



- The default constructor initializes a set to the so-called "empty set," i.e., a set for which all elements contain false.
- Provide member functions for the common set operations. For example, provide a unionOfSets member function that creates a third set that is the set-theoretic union of two existing sets (i.e., an element of the result is set to true if that element is true in either or both of the existing sets, and an element of the result is set to false if that element is false in each of the existing sets).
- Provide an intersectionOfSets member function which creates a third set which is the set-theoretic intersection of two existing sets (i.e., an element of the result is set to false if that element is false in either or both of the existing sets, and an element of the result is set to true if that element is true in each of the existing sets).
- Provide an insertElement member function that places a new integer k into a set by setting a[k] to true.
- Provide a printSet member function that prints a set as a list of numbers separated by spaces. Print only those elements that are present in the set (i.e., their position in the vector has a value of true). Print --- for an empty set.
- Now write a driver program to test your IntegerSet class. Instantiate several IntegerSet objects. Test that all your member functions work properly.
- Output

```
Enter set A:
Enter an element (-1 to end): 45
Enter an element (-1 to end): 76
Enter an element (-1 to end): 34
Enter an element (-1 to end): 6
Enter an element (-1 to end): -1
Entry complete

Enter set B:
Enter an element (-1 to end): 34
Enter an element (-1 to end): 8
Enter an element (-1 to end): 93
Enter an element (-1 to end): 45
Enter an element (-1 to end): -1
Entry complete

Union of A and B is:
{   6   8  34  45  76  93   }
Intersection of A and B is:
{  34  45   }

Inserting 77 into set A...
Set A is now:
{   6  34  45  76  77   }
```

Paste Code and Screenshot here

Question 4:

Write a University management system console application in C++ with the following functionality.

The program must have two classes **University** and **Department**

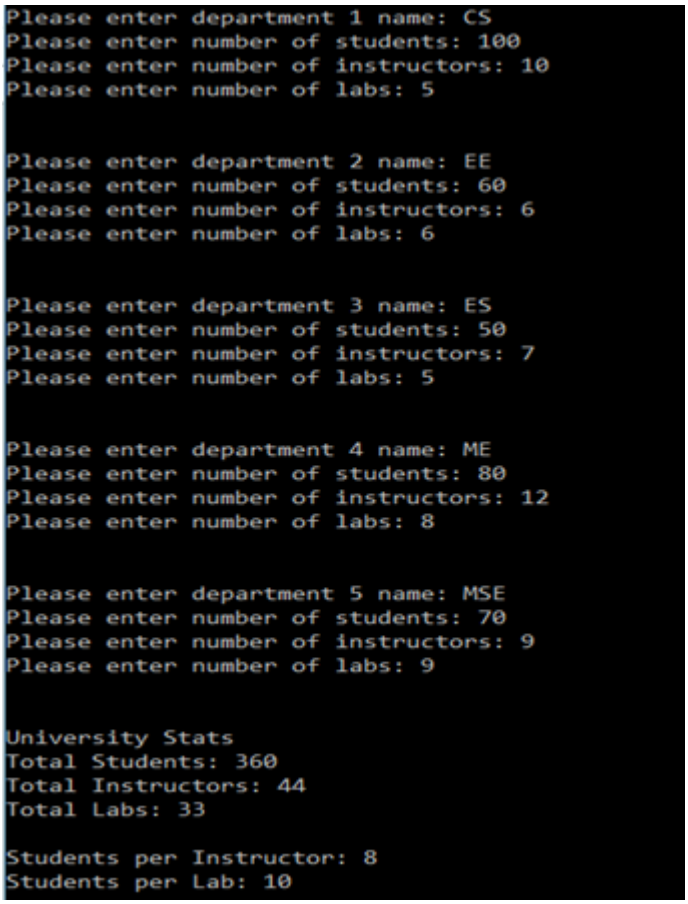
1. **Department Class:**
   Must have a name, number of students, number of instructors and number of labs as private data members
   Must have University as a friend class

2. **University Class:**
   Must have an array of 5 departments as private members.
   Must have a friend function named printStats.

printStats must display name, Total no. of students, Total no. of instructors, Total no. of labs, students per instructor and labs per instructor of the university on console screen.

The program must prompt the user to enter the Department Name, Number of Students, Number of Instructors and Number of Labs of the university.

The program must display above mentioned stats using friend function.



Paste Code and Screenshot here