Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

Instructions

1- The assignment is individual**.**
2- Deadline of submission is **Jan. 14th at 11:55 pm.**
3- Please submit only work that you did yourself. If you copy work from your friend or book or the internet (or give your work to another student), you will lose the marks of this assignment.

Application (16 Marks to be scaled to 4)

List - Stack - Queue

Stacks are based on the LIFO (last in- first out) principle, i.e., the element inserted at the last, is the first element to come out of the **list**.
Queues are based on the FIFO (first in- first out) principle, i.e., the element inserted at the first, is the first element to come out of the **list**.

Using inheritance, templates and exceptions, the team will develop base a **class template** MyList that stores an array of generic-typed elements **elems** with a capacity **itsSize.**

The class MyList is an abstract class which has the following:

- Parameterized constructor that takes the array size.
- A destructor
- getSize which returns the capacity of the array
- The pure virtual functions:
  - addElem which adds an element
  - getElem which returns an element, after removing it from the array.
  - isEmpty which returns true if the array is empty
  - isFull()which returns true if the array is full
  - clearItems() which empties the array;

The classes MyStack and MyQueue will inherit from MyList, and implement the pure virtual functions according to the definition of each above.
In MyStack the last element added will be the element returned if we call the getElem method.
In MyQueue the first element added will be the element returned if we call the getElem method.

Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

In main the user will be able to store **integers**, **strings** or **Rectangle objects** in our stack/queue based on the user's choice. The class Rectangle has the following specification:

```
class Rectangle
{
        int length;
        int width;
      public:
        Rectangle();
        Rectangle(int,int);
        int getLength();
        int getWidth();
        int getArea();
        friend ostream& operator<<(ostream&,const Rectangle&);
        //the overloaded function prints length, width & area of the rectangle
};
```

Before adding an element check that the stack/queue is **not full**, and before getting an element check that the stack/queue is **not empty**.
In main: **First,** ask the user whether to use stacks or queues (**S/Q**),
Second, ask about the number of elements to be added,
Then, ask whether the user wants to store integers, strings or rectangles (**1/2/3**),
Finally, loop for number of elements to take in the data to store, then display all the data.
In case rectangle object are stored, length, width and area are displayed. The stack should show the elements in reverse order, while a queue shows the elements the same order they were entered.

Exceptions:

Handle exceptions that might occur because of a wrong input from the user, **such as:**
User entered negative or zero number of elements,
User chose to store integers but entered a string value,
User chose to store Rectangles objects and entered a negative length or width (use exception classes defined in Rectangle class for this kind of exception)

Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

User Scenario 1:

```
Would you like to use a stack or a queue (S/Q)?
S
How many items to store?
3
Would you like to store integers, strings, or rectangles (1,2, or 3)?
1
Enter num:11
Enter num:22
Enter num:33

All Elements
Element-->33
Element-->22
Element-->11

Process returned 0 (0x0)   execution time : 7.123 s
Press any key to continue.
```

User Scenario 2:

```
Would you like to use a stack or a queue (S/Q)?
S
How many items to store?
2
Would you like to store integers, strings, or rectangles (1,2, or 3)?
1
Enter num:hello
Wrong String Input-->will enter 0
Enter num:44

All Elements
Element-->44
Element-->0
```

Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

User Scenario 3:

```
Would you like to use a stack or a queue (S/Q)?
s
How many items to store?
4
Would you like to store integers, strings, or rectangles (1,2, or 3)?
2
Enter string:object
Enter string:oriented
Enter string:programming
Enter string:cs213

All Elements
Element-->cs213
Element-->programming
Element-->oriented
Element-->object
```

User Scenario 4:

```
Would you like to use a stack or a queue (S/Q)?
S
How many items to store?
3
Would you like to store integers, strings, or rectangles (1,2, or 3)?
3
Enter length and width space separated:4 3
Enter length and width space separated:6 7
Enter length and width space separated:2 8

All Elements
Element-->Length=2,Width=8,Area=16

Element-->Length=6,Width=7,Area=42

Element-->Length=4,Width=3,Area=12
```

Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

User Scenario 5:

```
Would you like to use a stack or a queue (S/Q)?
Q
How many items to store?
4
Would you like to store integers, strings, or rectangles (1,2, or 3)?
2
Enter string:faculty
Enter string:of
Enter string:computers
Enter string:&artificial.intelligence

All Elements
Element-->faculty
Element-->of
Element-->computers
Element-->&artificial.intelligence
```

User Scenario 6:

```
Would you like to use a stack or a queue (S/Q)?
Q
How many items to store?
3
Would you like to store integers, strings, or rectangles (1,2, or 3)?
3
Enter length and width space separated:7 6
Enter length and width space separated:-1 7
Wrong length or width entering 0 and 0
Enter length and width space separated:4 5

All Elements
Element-->Length=7,Width=6,Area=42

Element-->Length=0,Width=0,Area=0

Element-->Length=4,Width=5,Area=20
```

Bonus(4 marks to be scaled to 1 mark):

Make your stack & queue resizeable such that no capacity is taken as input from the user and the user is asked to add 1 element at a time after being asked if (s)he wishes to enter another element. Modify **addElem** of the stack & queue such that it actually extends the memory allocated to the array by 1 element every time it is called.
Note that you will not need to call isFull before adding an element to the stack because the stack/queue is never full and you can always add more.

Object-Oriented Programming

Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

Sample Scenario for the Bonus:

```
Would you like to use a stack or a queue (S/Q)?
q
Would you like to store integers, strings, or rectangles (1,2, or 3)?
3
Enter length and width space separated:5 4
Would you like to enter another elem?(y/n)
y
Enter length and width space separated:9 8
Would you like to enter another elem?(y/n)
y
Enter length and width space separated:22 2
Would you like to enter another elem?(y/n)
y
Enter length and width space separated:4 13
Would you like to enter another elem?(y/n)
n

All Elements
Element-->Length=5,Width=4,Area=20

Element-->Length=9,Width=8,Area=72

Element-->Length=22,Width=2,Area=44

Element-->Length=4,Width=13,Area=52
```

General Notes:

1- **Submission will include <u>1 cpp file</u> with 4 classes and their implementation (Rectangle, MyList, MyStack, MyQueue) and a main function that provides the same experience as the screenshots provided above.**
2- **All submitted code will be checked for plagiarism and matching codes will take zeros regardless of who is the source and who is the copier.**
3- **Bonus grade will not be counted if the rest of the program is not working correctly.**

Grading criteria (Total 16 marks)

- MyList class (2 marks)
- MyStack class (4 mark)
- MyQueue class (4 mark)
- Rectangle class (2 mark)
- Main function (4 mark)

Object-Oriented Programming
Assignment 3 (16 Marks)

Cairo University, Faculty of Computers and Information

Submission Instructions

1- The student will submit **1 cpp file** named by your group number, an underscore, then ID. Ex. (**S3_20192222.cpp**)

2- **The student must understand the details** of the entire program and be able to explain it or even modify it if needed.

By failing to follow the submission instructions of the assignment, you are risking your assignment's grade.