Lab 05 - Stacks

Direction: Modify and submit the template provided to the Labs directory of your github repository and/or as an attachment on Google classroom under the accurate Lab05 assessment. You must not include additional libraries.

Part A: In class

Your objective is to write a complete program that defines a bool function named `ValidEnclosure()` whose header is

```
bool ValidEnclosure(string str)
```

It returns true if *str* represents a proper enclosure of parentheses, `()` and curly brackets, `{}`; otherwise, it returns false. If there are any other characters besides parentheses and curly brackets, the function should return false. For instance, the calls `ValidEnclosure("{}({})")` and `ValidEnclosure("({)}")` will return `true` and `false` respectively.

Part B: Take home

Your objective is to define a class named **MaxStack** which is a stack with an additional method that returns the maximum value in the stack. Your stack will be finite with a maximum capacity of 100 and its elements will be int values. Your class, must inherit the **StackInterface** interface in the `Interface.h` header file and it must include the following:

- ☐ A private int array field with a of size 100.
- ☐ A private int field that will be pivot *top*.
- ☐ A public default constructor that initializes *top*.
- ☐ A public copy constructor.
- ☐ A public assignment operator.
- ☐ A public empty destructor.
- ☐ A public overridden `Push()` method. It adds the parameter to the top of the stack if the stack is not full.
- ☐ A public overridden `Pop()` method. It removes the top element of the stack if the stack is not empty.
- ☐ A public bool constant method named `IsFull()` that takes no parameters. It returns true if the stack is full; otherwise, it returns false.
- ☐ A public overridden `IsEmpty()` method. It returns true if the stack is empty; otherwise, it returns false.
- ☐ A public overridden `Top()` method. It returns the top element of the stack if the stack is not empty; otherwise, it throws the error message `"Empty Stack"`.
- ☐ A public constant int reference constant method named `Max()` that takes no parameters. It returns the maximum value in the stack if the stack is not empty; otherwise, it throws the error message `"Empty Stack"`.
- ☐ A public constant method named `ToString()` that takes no parameters. It returns a string of the value of the top element and the maximum value in the stack enclosed in parentheses separated by a comma.
- ☐ A friend overloaded ostream operator. It displays the elements of the set in the same format as `ToString()`.

You can add additional fields if you find it necessary. **If you can make all the method have a constant big-O runtime except for the special member functions, you will receive extra credit.**