**Lab 02**

**Instructor:** Shahriar Ivan

**General instructions:**

Create a java application project naming the java file as Lab02\_2A\_ID where ID is your student ID. The example snippets will use the general class name Lab02\_2A without the ID portion. If there are multiple tasks, you don’t have to create separate projects for each task. A single project file should contain all the .java files that would be necessary to satisfy all the tasks given here.

**Tasks:**

Stack is a LIFO data structure. Java has its own built-in class for Stack. But for this lab, we will be creating our own implementation of Stack. So, the use of the built-in Stack class is forbidden for this task.

Now, within our project, we already have a public class with main function created by default. In addition to that, we can create another Java class within the same project and name this class as myStack. The basic structure of that new class is given below:

package lab02;  
public class myStack  
{  
 char items[]; *//private attribute* int top; *//private attribute  
  
 //Following is the constructor that initializes the attributes* myStack(int size)  
 {  
 items = new char[size];  
 top = -1;  
 }  
  
 public void push(char c) *//COMPLETE THIS PART*  
 {

*//should add a new element in the items array and update the top* }  
  
 public char pop() {  
 if(isEmpty())  
 {  
 System.out.println("Nothing left to pop");  
 System.exit(1);  
 }  
 *// COMPLETE THIS PART - should return the top element and reduce the top* }

public Boolean isEmpty() *// COMPLETE THIS PART*  
 { *// should return true or false based on whether the stack is empty*

}  
}

JAVA

Then we go back to the public class with the main function where we declare an object of the newly created myStack class. And we take in an input string of parentheses. The string of parenthesis may or may not be balanced. A string of parenthesis is said to be balanced if for every opening parenthesis there is a corresponding closing parenthesis in a symmetric order. For example, “(){}[]” is balanced, but “({)}[]” is not balanced. In the example, we take a balanced string:

public class Lab02\_2A  
{  
 public static void main(String[] args)  
 {  
 int size = 100;  
 myStack st = new myStack(size);  
 String brackets = "[{()}]";  
 *// COMPLETE THIS PART - Code for finding if the string of brackets is balanced or not* }  
}

JAVA

1. Use the example code as a starting point and complete the implementation of the myStack class by filling in the methods given in the example code.
2. Go to the public class with the main function and declare an object of myStack class as shown in the example code. Fill in the code that will determine whether the String object brackets contains a balanced string of parenthesis or not. The output of the main function should be “Balanced” if it is balanced, or “Not balanced” otherwise.