Homework Assignments 3
Due date: February 20, 11:55PM EST.

You do not have to write the actual programs for any of the problems, but you may want to experiment with some code to verify your own answers.

Problem 1

Write a method that given a String objects that contains a valid postfix expression, evaluates and returns its value. You can assume that the expression is valid (contains the right number of operators and operands in right places) and that it consists of only numerical values (integers only) and arithmetic operators (+, -, *, /).

Your method should use the algorithm provided in class:

Algorithm for evaluating postfix expressions

for each character in the input postfix string expression
    if the character is an operand
        push it (its value) onto a stack
    else if the character is an operator
        operand2 = pop stack
        operand1 = pop stack
        compute operand1 operator operand2
        push result onto stack
return top of the stack as result

You may use the Stack generic class provided by Java:
http://docs.oracle.com/javase/7/docs/api/java/util/Stack.html.
In order to create a stack of integers, for example, you would write

    Stack<Integer> stack = new Stack();

Hint: It might be helpful to construct a Scanner object from the String object passed to the class using Scanner’s constructor that takes String as a parameter.

Problem 2

Show the output of the following programs or state that they are not correct and why:
A)

```java
public class Test {
    public static void main(String[] args) {
        new Person().printPerson();
        new Student().printPerson();
    }
}

class Student extends Person {
    public String getInfo() {
        return "Student";
    }
}

class Person {
    public String getInfo() {
        return "Person";
    }
    public void printPerson() {
        System.out.println(getInfo());
    }
}
```

B)

```java
public class Test {
    public static void main(String[] args) {
        new Person().printPerson();
        new Student().printPerson();
    }
}

class Student extends Person {
    private String getInfo() {
        return "Student";
    }
}

class Person {
    private String getInfo() {
        return "Person";
    }
    public void printPerson() {
        System.out.println(getInfo());
    }
}
Problem 3

Given the CharQueue ADT below:

**Informal Specification**

The `IntegerQueue` contains a (possibly empty) collection of objects of type `Integer`. The queue supports the following operations:

- **insert**
  - This operation adds a `Integer` object, given as a parameter, to the end of the queue of integers.

- **remove**
  - This operation removes and returns a `Integer` object from the front of the queue of integers. If the queue is empty, `null` should be returned.

- **toString**
  - This operation produces a String that contains all `Integer` objects stored in the queue from front to back separated by a single comma and a single space. If the queue is empty, an empty string should be returned.
public interface IntegerQueue {
    public void insert ( Integer item );
    public Integer remove ( ) ;
    public String toString ( ) ;
}

State the return value and show the content of the, initially empty, queue of Integer objects after each of the following operations. If any of the operations below is invalid or would cause program to crash, state it and explain what is wrong with it (then proceed with the next step ignoring the invalid line). Assume that queue is a reference of type IntegerQueue.

queue.insert( 15 );
queue.insert( 3 );
queue.insert( -15 );
queue.insert( 35 );
queue.remove();
queue.toString();
queue.remove();
System.out.print(queue);
queue.remove();
queue.insert( 13 );
queue.toString();
queue.remove();
queue.remove();
queue.remove();
queue.insert( 3 );

Assume that array based queue implementation is used.

How and what to submit

You can use the template provided at https://docs.google.com/document/d/1aA4Pge0yqxLPDVWWUyVcHM0Fr1Ew_lGvmBmhSZo9QqM/edit?usp=sharing to complete your solution.

The completed solutions should be submitted as PDF documents to NYU Classes.