Write a class `Postfix` which uses a stack to evaluate a numerical postfix expression. Your class must contain the following method:

```java
public static double evaluate (String exp){
    which returns the double value of the postfix expression stored in the string parameter exp. Here is an outline of the method code:

    public static double evaluate (String exp){
        Stack<Double> s = new Stack<Double>();
        String[] tokens = exp.split("\s+"),
        for (int i=0; i<tokens.length; i++)
            // if tokens[i] is an operator +,-,*,/
            // then pop op2 and op1 off the stack s (in this order)
            // and push (op1 operator op2) onto the stack
            // fill in code here:

            // else (if tokens[i] is not an operator), it is a number
            // push the number value Double.valueOf(tokens[i])
            // onto the stack
            // fill in code here:

        }
        return s.pop();
    }

Your class may contain other methods (which must be static) as you see fit.

In the above code, the method `split` of the `String` class splits the `String exp` into an array of `Strings` separated by spaces (to use spaces as delimiters in the `split` method is encoded in the parameter "\s+"

which is an example what is known as a Java regular expression).
2. Driver file

Your code should work with the following driver class:

```java
import java.util.Scanner;
public class Driver{
    public static void main (String[] args){
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter a postfix expression, with all "
            + "numbers and operators separated\nby "
            + "spaces (or q to quit):\n");
        String exp = scan.nextLine();
        while (! exp.equals("q") && ! exp.equals("Q")){
            double value = Postfix.evaluate(exp);
            System.out.println("The value is "+value+".");
            System.out.println ("Enter a postfix expression, with all "
                + "numbers and operators separated\nby "
                + "spaces (or q to quit):\n");
            exp = scan.nextLine();
        } // end while
    } // end main
} // end Driver
```

You may assume the user enters a valid postfix expression, with all the numbers and operators separated by one or more spaces.
3. Sample output

Assume the user enters at the prompt the following lines:

```
6 7 * 13 +
7 5 -10.2 13 - + / 15 .3 * *
17 3 -
5.2
q
```

Then the following should be on the screen:

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
6 7 * 13 +
The value is 55.0.
Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
7 5 -10.2 13 - + / 15 .3 * *
The value is -1.7307692307692308.
Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
17 3 -
The value is 14.0.
Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
5.2
The value is 5.2.
Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
q

There may be various exceptions thrown if the user enters expressions which are not valid postfix expressions (including those in which the numbers and operators are not separated by spaces).

4. Due date

Wednesday, March 31, 2010.

5. How to turn it in

Email your file Postfix.java to loftin@rutgers.edu as an attachment. Be sure to put your name in a comment line at the beginning of your file.
6. Bonus

Implement exception handling for the class Postfix. In particular, use the following exception class

```java
public class PostfixException extends RuntimeException{
    public PostfixException(String s){
        super (s);
    }
}
```

Your code should throw a `PostfixException` in the following circumstances:

- When the empty string is passed into the `evaluate` method.
- When, in your `evaluate` method, any attempt is made to pop from an empty stack. Use the `empty` method to check this.
- When, at the end of your `evaluate` method, there is more than one item on the stack. (There should be exactly one item, which is the return value; if there are more, it is not a valid postfix expression, as there should be more operators to combine the remaining numbers on the stack.)
- When an invalid token (neither a number nor an operator) is entered as part of a postfix expression. This includes the case when one or more valid tokens are not separated by spaces. For example, in the expression `3 x +`, the token `x` is invalid. Also, for the expression `3 5 6+ *`, the token `6+` is invalid. With each such exception, include an appropriate error message detailing which token is problematic.

To do this, you should check to see if each string `tokens[i]` is either one of the four valid operators or is a valid Java number. One way to check if `tokens[i]` is a valid number is to use the fact that `Double.valueOf` method throws a `NumberFormatException` if the parameter is not a valid Java number. Then your program should catch this exception and rethrow it as a `PostfixException` with the appropriate error message.

Your goal should be to ensure that the only exceptions ever thrown by the `Postfix` class are of class `PostfixException`. 
Here is a sample output

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
-2 6 3 + *
The value is -18.0.

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
PostfixException: invalid token:
    at Postfix.evaluate(Postfix.java:46)
    at Driver.main(Driver.java:15)

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
3 / 9
PostfixException: invalid postfix expression
    at Postfix.evaluate(Postfix.java:22)
    at Driver.main(Driver.java:15)

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
6 7 8 /
PostfixException: invalid postfix expression
    at Postfix.evaluate(Postfix.java:62)
    at Driver.main(Driver.java:15)

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
19.2 3 /
The value is 6.3999999999999995.

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
12 5 7+ *
PostfixException: invalid token: 7+
    at Postfix.evaluate(Postfix.java:46)
    at Driver.main(Driver.java:15)

Enter a postfix expression, with all numbers and operators separated by spaces (or q to quit):
q
using the modified driver class

```java
import java.util.Scanner;

public class Driver{
    public static void main (String[] args){
        Scanner scan = new Scanner (System.in);
        System.out.println("Enter a postfix expression, with all "
                          + "numbers and operators separated\nby "
                          + "spaces (or q to quit): ");
        String exp = scan.nextLine();
        while (! exp.equals("q") && ! exp.equals("Q")){
            try{
                double value = Postfix.evaluate(exp);
                System.out.println("The value is " + value + ".");
            }
            catch (PostfixException e){
                e.printStackTrace();
            }
        System.out.println("Enter a postfix expression, with all "
                          + "numbers and operators separated\nby "
                          + "spaces (or q to quit): ");
            exp = scan.nextLine();
        } // end while
    } // end main
} // end Driver
```