CSCI 1101
Computer Science II

Assignment No. 5
Date Given: March 29, 2013
Date Due: April 8, 2013

This assignment is on conversion from infix to postfix and evaluation of postfix expressions. The objective is as follows:

Input to the program:
```
( 5.0  +  3.0 )  *  5.0  –  2.5
```

Output:
The postfix expression is 5.0 3.0 + 5.0 * 2.5 –
The expression evaluates to 37.5

Here are the steps:
1. Read the input as a string. ***You may assume that there is at least one space between each number, bracket or operator.***

2. ***Tokenize the string*** by breaking it into its individual components. Tokenizing strings can be done using a simple code that uses the StringTokenizer class. Here's an example program. It reads a string like the one given above and prints the individual components that are separated by spaces.

```
import java.util.Scanner;
import java.util.StringTokenizer;
public class StringTokenizerExample
{
      public static void main(String[] args)
      {
            Scanner keyboard = new Scanner(System.in);
            String input = keyboard.nextLine();
            String component;

            StringTokenizer str = new StringTokenizer(input, " ");
            while (str.hasMoreTokens())
            {
                  component = str.nextToken();
                  System.out.println(component);
            }
      }
}
```

For the input:
```
( 5.0 + 3.5 ) * 5.0 - 2.5
```

the following is the output:
```
(
5.0
+
3.5
)
*
5.0
-
2.5
```

3. Using the algorithm for conversion from infix to postfix discussed in the lectures, convert the input expression to postfix. For this, create a Stack that can hold Strings. You may use a Stack implementation based on arrays, arraylists or linkedlists.

4. Then using the algorithm for evaluation of postfix, evaluate the expression. Again use a Stack of Strings. You can use any Stack implementation. Note that when you evaluate the expression, you need to convert the String to Double for evaluation and back to String when pushing it into the Stack.

```
double d;
String str;
d = Double.parseDouble(str);        //converts string to double
str = Double.toString(d);           //converts double to string
```

Here's the overall structure of your solution:

```
import java.util.Scanner;
import java.util.StringTokenizer;
public class Assignment5
{
        public static void main(String[] args)
        {
                //declare variables

                //read the input infix expression

                //tokenize it and extract the individual components

                //as you extract the components use the infix to postfix conversion algorithm and
                //convert to postfix

                //print the postfix string

                //read the postfix string and evaluate it

                //print the answer

        }
}
```