CS 158 Lab 4 January 26, 2010
Postfix Calculator

1. As we discussed in class yesterday, here is the algorithm for the postfix calculator:
   - Grab input string and put it into a Scanner object
   - While scanner has a next token

```
token = scanner.next()
if ( isOperand(token) )
        stack.push(token)
else
        op2 = stack.pop()
        op1 = stack.pop()
        result = apply(token)
        stack.push(result)
```

   - Answer will be on top of the stack

   Create a new class in your Stacks project from last time called **Postfix** that contains a main method. Add an **ArrayStack**. We will not have any **Postfix** objects, so make all of your methods static. Don't forget to instantiate an actual stack in the main method.

2. Write the **isOperand** method so that it recognizes addition, subtraction, multiplication, division, and exponentiation as binary operators, and assumes that anything else is an operand. Write a JUnit test for this method and make sure the test passes.

3. Write the **apply** method so that it takes three arguments: the operator, and the two operands. You need to decide whether this method will return a String or a double. Either can be made to work. Once you decide, write some tests for the method, implement it, and make sure the tests pass.

4. Fill in the main method so that the user is prompted for an input string in postfix notation and the algorithm in part 1 above is implemented. You may add other methods to help as needed. Keep in mind that you may have to modify this program later. The program should print the answer in the form: **Result =** *whatever*

5. Notice that if you enter an invalid postfix string such as "3 4 5 +", the program currently computes an answer to be 9.0. Fix this so that an appropriate error message is printed instead.

6. Modify your program so that if the user enters an invalid value (either operator or operand), an appropriate error message occurs.

Copy your **Postfix** and main and test classes to the shared drive in your Stacks folder by Thursday morning before lab.