

Week 15 Lab –Stack Manipulation

Maximum Points = 10

Sometimes it's useful to define operations on an ADT without changing the type definition itself. For example, you might want to print the elements in a stack without actually adding a method to the Stack ADT (you may not even have access to it). To explore this, use either the Stack class provided by Java (in java.util) or one of the stack classes that you wrote in an earlier lab exercise and the test program *StackTest.java*. Add the following static methods to the StackTest class (the signature for these methods and the declaration in StackTest assumes you are using a stack class named Stack—modify them to use the name of your class):

- void printStack(Stack s)—prints the elements in stack s from top to bottom. When printStack returns, s should be unchanged.
- Stack reverseStack(Stack s)—returns a new stack whose elements are backwards from those in s. Again, s is unchanged.

If time permits, also do this method -

- Stack removeElement(Stack s, int val)—returns a new stack whose elements are the same as those in s (and in the same order) except that all occurrences of val have been removed. Again, s is unchanged.

Modify the main method to test these methods. Be sure you print enough information to see if they're working!

```
// *****  
// StackTest.java  
//  
// A simple driver to test a stack.  
//  
// *****  
import java.util.Stack;  
public class StackTest  
{  
    public static void main(String[] args)  
    {  
        // Declare and instantiate a stack  
        Stack stack = new Stack();  
  
        //push some stuff on the stack  
        for (int i=0; i<10; i++)  
            stack.push(i);  
        stack.push(5);  
  
        // call printStack to print the stack  
  
        // call reverseStack to reverse the stack  
  
        // call printStack to print the stack again  
  
        // call removeElement to remove all occurrences of the value 5 – save the  
        // stack returned from this call  
  
        // call printStack to print the original stack and the new stack.  
  
    }  
    static void printStack(Stack s)
```

```
// method that prints a stack
{
}

static Stack reverseStack(Stack s)
// method that reverses a stack
{

}
}
```

(Due before end of the day on Friday, December 3, 2010) Submit your .java files containing your program to the dropbox in WebCT.

Grades are determined using the following scale:

- Runs correctly.....:___/3
- Correct output.....:___/2
- Design of output.....:___/1
- Design of logic.....:___/2
- Standards.....:___/1
- Documentation.....:___/1