CSC 205 Lab 11 : Stacks

Goals

After completing this lab, you should be able to:

- Understand and be able to use the methods of the Stack ADT.
- Know some of the general applications of stacks.
- Be able to write simple methods to count the number of elements in a stack and print out a stack using both class methods and instance methods
- Understand how to implement a stack using a linked list.

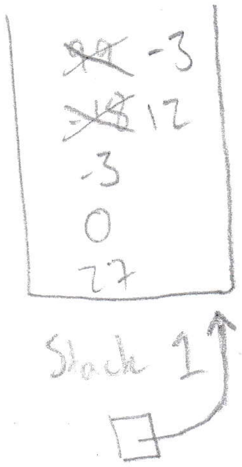
Lab Startup

Change into your `Labs` directory, and let's create and change into a `Lab11` directory. Now, let's copy over some files by typing : `cp /pub/digh/CSC205/Lab11/* .`

Building and Tracing a Stack

Draw the stack that would be created following the code segment below. Use your Stack ADT handout as a guide.

```
Stack stack1 = new Stack();
stack1.push(new Integer(27));
stack1.push(new Integer(0));
stack1.push(new Integer(-3));
stack1.push(new Integer(-18));
stack1.push(new Integer(99));
stack1.pop();
stack1.pop();
int x = ((Integer)stack1.top()).intValue();
stack1.push (new Integer(-12));
stack1.push (new Integer(x));
```



Now, compile and run the `MyStack` program to check your results.

Writing Simple Class Methods & Instance Methods with Stacks

Add a value-returning class method named `findMax` to your `MaxTest` program that can be used to return the largest `String` value stored in a stack of string values. By largest, we mean last in lexicographic (dictionary) order. Your method will have one parameter, a stack object, and should not modify this parameter. **You are writing this method assuming you are a user of the Stack class.** Here, you'll merely be using the