SYSC 2100, Winter 2013
Assignment 3: Stacks
Due: February 28, 2013
(Due at noon that day)

In the first assignment, you implemented a recursive method to determine the number of distinct ways in which a given amount of money in cents can be changed into quarters, dimes, nickels, and pennies. As discussed in class, we can replace recursion with the use of a `Stack`. So in this assignment, implement an ADT Stack and use it to re-implement the task from Assignment 1.

To do this, proceed in two steps. The first step is to implement and test your stack ADT. You can download an interface definition of the ADT (*PureStack.java*) from the course website. In this assignment, you are to

1. Implement a *LinkedListPureStack* class and test it
2. Use the *LinkedListPureStack* to implement a solution that typically requires recursion.

1) Implementing and Testing *LinkedListPureStack*

Provide the *LinkedListPureStack* class implementation of the *PureStack* interface, based on the use of a *LinkedList*. Test your implementation with a program of your choice, similar to the one you wrote to test the Linked List in Assignment 2. You do not need to submit this program though.

2) Using *LinkedListPureStack*

Re-implement either your solution or the posted sample solution for Assignment 1. Your implementation should not use recursion, rather operate on one (or multiple) instances of your *LinkedListPureStack* class.

Submission Requirements: Submit your assignment (the source files) using WebCT. Your program(s) should compile and run as is in the default lab environment, and the code should be well documented. Submit all Java class files without using any archive or compression as separate files. Your stack implementation should be submitted as file *LinkedListPureStack .java*, the main program should be saved in file *StackChangeMain.java*, if you need to define additional classes etc., you are free to name them according to your own needs. But the TA(s) should be able to run your application by entering **`java StackChangeMain`** on a command-line.

Marks will be based on:

- Completeness of your submission
- Correct solution to the problem
- Following good coding style
- Sufficient and high-quality in-line comments
- Adhering to the submission requirements

The due date is based on the time of the WebCT server and will be strictly enforced. If you are concerned about missing the deadline, here is a tip: multiple submissions are allowed. So you can always submit a (partial) solution early, and resubmit an improved solution later. This way, you will reduce the risk of running late, for whatever reason (slow computers/networks, unsynchronized clocks, failure of the Internet connection at home, etc.).
In WebCT6, you can manage the submission until the deadline, taking it back, deleting/adding files, etc, and resubmitting it. The system also provides online feedback whether you submitted something for an assignment. It may take a while to learn the submission process, so I would encourage you to experiment with it early and contact the TA(s) in case you have problems, as only assignments properly and timely submitted using WebCT will be marked and will earn you assignment credits.