Assignment Prefix: Lab05

Due Date: Tuesday, Oct. 3rd @ 11:59pm

Points: 100

This is an individual assignment.

**Restrictions:**

**You cannot use any predefined Java classes in writing this lab.**

**Create a NetBeans project named Lab05 and save it to a location like the desktop or your flash drive. In the project you will do the following:**

In this assignment you are to implement the Stack, Queue, and Deque structures describe in Chapter 6 of the textbook.

Begin this assignment by create a Netbeans project named Lab105 and copy the SinglyLinkedList class from a previous assignment into the project directory.

Then:

* Implement the Stack interface found in Code Fragment 6.1
* Implement the Array Stack class found in Code Fragment 6.2
* Implement the LinkedStack class found in Code Fragment 6.4
  + Note that the LinkedStack is implemented using the adapter design pattern
  + Be sure that you understand how an adapter design pattern works
* Implement the Queue interface found in Code Fragment 6.9
* Implement the ArrayQueue found in Code Fragment 6.10
* Implement the LinkedQueue found in Code Fragment 6.11
  + Again note the used of the adapter design pattern

In your client class you are going to compare the run times of the static (array based) classes to the dynamic (linked based) classes.

The basic test will be to:

* Start a timer
* Fill a queue with N elements
* Dequeue all of the elements from the queue onto a stack
* Pop all of the elements from the stack back into the queue
  + Note that this reverses the order of the queue
* Stop the timer.

You will perform two tests. One with array based stacks and queues and one with linked based stacks and queues.

Use random Integers as the element and start with N = one million.

Increase N by a factor or 10 until you would run out of memory or until the run time becomes excessive (more than 10 minutes). There may be other types of exceptions that you will need to deal with. In any case your program should exit gracefully and not crash. You cannot code a hard upper limit for N or time into your Client. You program needs to predict when it will fail and not run that test.

You client should be loop driven so that N is automatically increment on each pass.

You output should look something like (make sure that you line up as a nicely formatted table (hint, use printf, and in your Word Document use Courier New font):

N Array Linked

1,000,000 7 14

10,000,000 700 140

100,000,000 7,000 1,400

1,000,000,000 Out of Memory 14,000

10,000,000,000 Out of Memory Out of Time

Note that I just made up the above results off the top of my head. Your results will probably look different.

**Things to turn in:**

* Open a Microsoft Word document
* Copy and Paste the source code of the all your classes (make sure to use   
  *Ctrl + A* to select all the source code of the program, *Ctrl + C* to copy, and Ctrl + V to paste.).
* Copy and Paste the source code of the **Client** Class
* Copy and paste the output of the client program
* Next, zip the Project folder.
* Finally on blackboard, submit both your Word document and project zipped file.