**ASSIGNMENT TWO**

For this assignment, you are to implement 2 generic Stacks; one backed by an array, and other backed with a linked chain of objects (i.e., Node).  You should implement the attached interface (Stack) and call your classes ArrayStack and LinkedStack.  I have attached a JUnit test class that will test your implementations and output some performance measurements.

Grading:

* + - 25 points: your LinkedStack and ArrayStack both implement the specified interface and compile.
    - 3 points: your code passes the testPeek test.
    - 4 points: your code passes the testPopAndPush test.
    - 3 points: your code completes the measurement tests.
    - 5 points: style / implementation / grader's discretion.

10 points:

Run the tests a few times and look at the measurement results. Answer the following questions in a .txt or .doc file:

* + 1. Which implementation is faster for small amounts of pushes?
    2. Which implementation is faster for small amounts of pops?
    3. Which implementation is faster for larger amounts of pushes?
    4. Which implementation is faster for larger amounts of pops?
    5. How do you explain your results for 1-4?  Were any results surprising?  Why or why not?
    6. What are the tradeoffs to be considered in these implementations?

Please submit files named ArrayStack.java and LinkedStack here by September 27 at noon (eastern standard time).  You do not need to submit the JUnit test file or the interface.

*Note: unfortunately as we have seen, Java does not allow you to instantiate a generic array.  To get around this, the simplest approach is to use an array of Object (i.e., Object[] ) and cast any return values to T.*

*Note: remember to declare your classes as "public class ArrayStack implements Stack " or public class LinkedStack implements Stack " or you may get odd compile errors.*