STACKS LAB NOTES

UMD CS DEPARTMENT

1. Overview

In this Lab, you will implement an abbreviated version of Java's `Stack` interface. Besides requiring fewer operations, your implementation will not be "generic" in that you will not be able to specify your own type information at definition time. In other words, Java's `Stack<T>` allows

```
public static void main( String[] args ) {
    Stack<String> names = new Stack<String>();

    names.push("TomR");
    String nameOnStack = names.pop();
    ...
}
```

whereas your stack will assume only `Object` types. This means that you (as the user of your `Stack` class) will need to manage type conversions (type casting) when retrieving or peeking objects out of the `Stack` objects that you create.

```
public static void main( String[] args ) {
    Stack names = new Stack();

    names.push("TomR");
    String nameOnStack = (String)names.pop();
 ...
 }
```

2. Procedure

You will note that one file is named `IStack.java`; it contains the definition of the `Stack` interface. You need to do nothing with this file other than read the documentation contained therein.

Your work will focus on the `MyStack.java` file, where you will use a "backing array" in order to implement the `Stack` interface. Your Teaching Assistants will help you with this.

Use the documentation for the `Stack<E>` class in the online Java documentation as your guide as to what these methods need to do. Don't forget to implement your own `StudentTests`.