All code for this assignment should be in package edu.syr.hw1. Follow the same convention for future assignments, i.e. code for homework 2 should be in package edu.syr.hw2 and so on.

For the most part your submissions will be graded by running unit tests. Don't change the structure of any code given to you as part of the assignment. More specifically, class names, method signatures, field visibility should not be changed. Check with the TA for specific questions.

1. Write a class, Hello, which has a main method which prints "Hello World" to stdout. This class will be run like so: java -cp . edu/syr/hw1/Hello

2. Write a class, Greeting, which has a method greet(), which prints "Hello World" to stdout. This class will be run like so:

```
import syr.edu.hw1.Greeting

public class Runner {
   public static void main(String[] args) {
      Greeting g = new Greeting();
      g.greet();
   }
}
```

3. Write a class, Library, with init() and search() methods.

The init() method will be passed a String[] as parameter. An example of the String in the array:
"The Go Programming Language, Alan Donovan and Brian Kernighan"

The init method should keep a reference to the array param in a field. This list is the catlog of the Library, i.e. these are all the publications in the Library available to be borrowed.

The search() method will be passed a String as parameter, and will return one of the matching items stored in the Library's catalog.

4. Complete the implementation of the IntMatrix class.

The idea is to use a one-dimensional array as a matrix. The field data should be initialized by the constructor to a size large enough to hold all the elements of the matrix. Feel free to add other fields as needed.

The get() and set() methods should convert the row and column parameters to

an appropriate index, and do error checking based on the size of the matrix.

For example:

```
IntMatrix m1 = new IntMatrix(2, 3);
// data should be initialized to int[6]
// and should be all 0s: {0, 0, 0, 0, 0, 0}
m1.set(0, 0, 1);
m1.set(0, 1, 2);
m1.set(0, 2, 3);
// m1 is now {{1, 2, 3}, {0, 0, 0}}
// data is now {1, 2, 3, 0, 0, 0}
m1.set(0, 3, 4); // throws exception; invalid cell
m1.set(1, 0, 4);
m1.set(1, 1, 5);
m1.set(1, 2, 6);
// m1 is now {{1, 2, 3}, {4, 5, 6}}
// data is now {1, 2, 3, 4, 5, 6}
```