

COS 126	General Computer Science	Spring 2009
<b>Programming Exam 2</b>		

This test has 2 parts. You have 50 minutes. The exam is open book, open note, and open web. You may use code from your programming assignments or the *Introduction to Programming in Java* booksite. No communication with any non-staff members is permitted. Submit your solution via Moodle. **Write out and sign the Honor Code pledge before turning in the test.**

*“I pledge my honor that I have not violated the Honor Code during this examination.”*

**Name:**

-----

**Signature**

**NetID:**

Part	Score
1	
2	
Total	

- P01 TTh 1:30 Will
- P01A TTh 1:30 Rob
- P01B TTh 1:30 Aditya
- P01C TTh 1:30 Michael
- P02 TTh 2:30 Will
- P03 TTh 3:30 Rob
- P04 TTh 7:30 Chris
- P05 WF 10 JP
- P06 WF 1:30 Chris
- P06A WF 1:30 Thomas
- P06B WF 1:30 Donna
- P06C WF 1:30 Michael

**Part 1.** Write a data type `Location.java` that models a named location on the surface of the earth (specified by its latitude and longitude in degrees) by implementing the following API:

```
public class Location
-----
    // create a new location with the given name and coordinates
    public Location(String name, double latitude, double longitude)

    // return the great-circle distance between this location and that
    public double distanceTo(Location that)

    // return a string representation of this location
    public String toString()
```

**Geometry.** The great-circle distance between two locations  $(x_1, y_1)$  and  $(x_2, y_2)$  is:

$$distance = 1.15077945 \times 60 \times \arccos(\sin x_1 \sin x_2 + \cos x_1 \cos x_2 \cos(y_1 - y_2))$$

Note that this formula assumes that the (latitude, longitude) values are in degrees (whereas Java's trigonometric functions use radians) and it computes the result in *statute miles*.

**Test client.** To test your implementation, you may use the client

```
public class LocationTest {
    public static void main(String[] args) {
        Location loc1 = new Location("PRINCETON_NJ", 40.366633, 74.640832);
        Location loc2 = new Location("ITHACA_NY", 42.443087, 76.488707);
        double distance = loc1.distanceTo(loc2);
        System.out.printf("%6.3f miles from\n", distance);
        System.out.println(loc1 + " to " + loc2);
    }
}
```

which is available at:

<http://www.cs.princeton.edu/introcs/data/LocationTest.java>

It should behave as follows:

```
% java LocationTest
172.367 miles from
PRINCETON_NJ (40.366633, 74.640832) to ITHACA_NY (42.443087, 76.488707)
```

**Part 2.** Write a client program `Postal.java` that takes the name of a ZIP code file (see below for file format) as a command-line argument, reads the data from the file, and stores it in a symbol table. Then, repeatedly read pairs of ZIP codes from standard input and output the great-circle distance between them (in statute miles). This distance is used by the post office to calculate shipping rates.

```
% java Postal zips.txt
08540 14853
172.367 miles from
PRINCETON_NJ (40.366633, 74.640832) to ITHACA_NY (42.443087, 76.488707)

08540 19072
40.558 miles from
PRINCETON_NJ (40.366633, 74.640832) to NARBERTH_PA (40.01768, 75.2594)
```

**ZIP code data file format.** The ZIP code data file consists of a sequence of lines, each line containing the ZIP code (a `String`), post office name (a `String`), latitude (a `double`, in degrees), and longitude (a `double`, in degrees), separated by whitespace. To test your client, you may use the file `zips.txt`

```
% more zips.txt
42601 AARON_KY          36.812827  85.199114
16820 AARONSBURG_PA    40.876944  77.387977
31794 ABAC_GA          31.451722  83.498867
...
14853 ITHACA_NY        42.443087  76.488707
...
08540 PRINCETON_NJ     40.366633  74.640832
...
71486 ZWOLLE_LA        31.613790  93.663569
```

which is available at:

<http://www.cs.princeton.edu/introcs/data/zips.txt>

**Submission.** Submit `Location.java` and `Postal.java` via Moodle. You do not need to submit `StdIn.java`, `In.java` or `ST.java`.

**Grading.** *Your program will be graded on correctness, clarity (including comments), design, and efficiency. You will lose a substantial number of points if your program does not compile or if it crashes on typical inputs.*