

More Practise with Expressions

Purpose:

To demonstrate an understanding of basic input and output in JAVA.

Due Date:

The completed lab assignment is due Friday, September 28 *at the beginning of lecture*. It should be stapled and your name highlighted.

Question and Answer Problems

1. The *great circle distance* between two points on the surface of the earth is essentially the shortest distance between the points (say by airplane). It can be calculated by the following formulas:

$$\begin{aligned}
 R &= \text{the radius of the earth} && = 6378.10 \text{ km} \\
 \phi_1 &= \text{the latitude of the first place} \\
 \theta_1 &= \text{the longitude of the first place} \\
 \phi_2 &= \text{the latitude of the second place} \\
 \theta_2 &= \text{the longitude of the second place} \\
 \phi_3 &= \phi_1, \text{ in radian,} && \theta_3 = \theta_1, \text{ in radian} \\
 \phi_4 &= \phi_2, \text{ in radian,} && \theta_4 = \theta_2, \text{ in radian} \\
 z_1 &= \cos \phi_3 && z_2 = \cos \phi_4 \\
 x_1 &= \sin \phi_3 && x_2 = (\sin \phi_4) \cos(\theta_4 - \theta_3) \\
 a &= x_1 x_2 + z_1 z_2 \\
 D &= R \cos^{-1}(a) = \text{the great circle distance.}
 \end{aligned}$$

The latitudes ϕ_1 and ϕ_2 should be in decimal degrees with negative numbers standing for South latitudes. The longitudes θ_1 and θ_2 should be in decimal degrees with negative numbers standing for West longitudes. The remaining angles are in radians. All of the trigonometry functions can be found in the `Math` class, including a function to convert from degrees to radians (although multiplying by $\pi/180$ also works). To find the names of the `Math` functions look in <http://download.oracle.com/javase/6/docs/api/>, or start from my Java pages.

Write a program that produces output similar to that shown in Figure 1 on the following page. Here are some test points that you can use:

City	Longitude	Latitude
Prince George	122.766W	53.916N
Vancouver	123.13 W	49.256N
Paris	2.333E	48.866N

The distance from Prince George to Paris is approximately 9780 km.

Hints

- to get nicely displayed numbers you may want to use the `String.format` function (there's a lot of solutions in JAVA).
- To choose between the letters E and W, or between N and S, you may want to use ideas from Chapter 3.

Other Problems

2. Write a program to determine the precise rules used by the `%` operator when negative numbers are involved. That is, determine what expressions like `(-8) % (-3)`, `(-8) % 3`, and `8 % (-3)` compute. If possible state the precise rule used by JAVA (either in comments, or in output from your program).
3. We have talked about the binary representation of numbers, but have not yet used this information. There are four operators that work directly with the bits of integers:

<code>~</code>	not	unary. changes 1's to 0.s and vice versa.
<code> </code>	or	binary. yields 1 where either argument bit is 1.
<code>&</code>	and	binary. yields 1 where both argument bits are 1.
<code>^</code>	exclusive or	binary. yields 1 where exactly one argument bits is 1.

These operators work *bitwise*. For instance, `2|1` is 3, because (the last bits of) 2 are 10 and of 1 are 01. On the other hand, `2&1` is 0.

Write a program to determine the values of `(13 | 11)` • `(13 & 11)` • `(13 ^ 11)` • `(~ 6)`, and print them out. Are they what you expected?

```
cs100/lab2/1: java GreatCircleDistance
This program computes the great circle distance
between two points on the earth, given their latitudes
and longitudes in decimal degrees.
```

```
(use - for western longitudes and southern latitudes)
```

```
Latitude of point 1 in (decimal) degrees? 53.916667
Longitude of point 1 in (decimal) degrees? -122.766667
Latitude of point 2 in (decimal) degrees? 48.86
Longitude of point 2 in (decimal) degrees? 2.333
```

```
The great circle distance between
(53.92 N , -122.77 E) and
(48.86 N , 2.33 E)
is 9779.36 km.
```

Figure 1: Sample Great Circle distance calculations
