

# CS205 C/ C++ Programming - Assignment 1

You are asked to write a simple program that, as you will see, may not be as simple to write in C as it looks if you want to write robust programs. It will allow you to learn about basic input/output.

This program must prompt the user for the name of a first city, then its latitude and longitude, then for the name of a second city with its latitude and longitude, then compute the flying distance between the two and display. For example,

**The first city: <>**

**The second city: <>**

**The distance between <first city> and <second city> is <result> km**

Here is the formula for computing the distance (adapted from mathforum.org, provided by Doctor Rob):

Assume the Earth is a perfect sphere. Let all angles be measured in signed degrees (negative latitude means South; negative longitude means West).

$$\text{phi} = 90 - \text{latitude}$$

The North Pole has  $\text{phi} = 0$ , the South Pole has  $\text{phi} = 180$ , and  $0 \leq \text{phi} \leq 180$ .

$$\text{theta} = \text{longitude}$$

Greenwich, England, has  $\text{theta} = 0$ , and  $-180 \leq \text{theta} \leq 180$ .

Let the angles for the two points be  $(\text{phi1}, \text{theta1})$  and  $(\text{phi2}, \text{theta2})$ . Then compute

$$c = \sin(\text{phi1}) * \sin(\text{phi2}) * \cos(\text{theta1} - \text{theta2}) + \cos(\text{phi1}) * \cos(\text{phi2})$$

**Note: phi and theta should be in radians.**

Then the shortest great circle distance between the two points is

$$d = R * \arccos(c)$$

where R is the radius of the earth in kilometers, and the arccosine is taken between 0 and 180 degrees, inclusive. Earth radius: 6,371 km

**Some cities for testing:**

city	latitude	longitude
------	----------	-----------

Shenzhen	22.55	114.1
Beijing	39.9139	116.3917
New York, USA	40.7127	-74.0059
San Francisco, USA	37.7833	-122.4167
London, UK	51.5072	-0.1275
Paris, France	48.8567	2.3508
Kolkata, India	22.567	88.367
Moscow, Russia	55.7500	37.6167
Rio de Janeiro, Brazil	-22.9083	-43.1964
Sydney, Australia	-33.865	151.209444

For checking out if your results are roughly correct:

[http://www.webflyer.com/travel/mileage\\_calculator/](http://www.webflyer.com/travel/mileage_calculator/)