24-780 B—ENGINEERING COMPUTATION

Assigned: Mon. Aug. 30, 2021
Due: Tues. Sept. 7, 2021, 11:59pm

Problem Set 1: Great Circle Distance

Before we can do any programming, you must install the computer resources necessary. Follow the instructions given in Lecture 1 (see Piazza or Canvas) to get the integrated developing environment (IDE) that best suits you. Although I *highly recommend* Visual Studio, you are free to select anything that makes you comfortable. This course is about programming, not about compiling. That said, don't be a hero by picking the most basic/raw application.

Also, note that even if you use a Mac, installing a Windows simulator (Bootcamp, Parallels, etc.) to run Visual Studio and other Windows programs (e.g., SolidWorks, Ansys, etc.) is probably a good idea. Visual Studio is also available on [Virtual Andrew](#) so that you don't have to install it at all, although Virtual Andrew sometimes feels a bit slow.

Now for the real task:

A couple of years ago, I was planning a trip in my electric car from Pittsburgh to Toronto. I noticed in my charging network app that there was a fast charger near Niagara Falls that was 220 miles away, well within reach of the battery's range of about 280 miles at high speed. With this information, I set off on my trip with peace of mind that I would not be stuck somewhere in the middle of nowhere with a depleted battery. As I was driving away from Pittsburgh, about 50 miles into the trip, it occurred to me that the app, instead of showing me a route, it had a link to a mapping app. This implied that 220 miles was the "straight-line" distance, not the "by-the-road" distance, which was actually around 280 miles. Fortunately, there was another charger a bit closer and this time I did use the mapping app to find that it was 190 miles from Pittsburgh, so I was able to do the trip with just a small change to my route.

The whole experience left me with the following thought:

> If the charging network app could not figure out the real travel distance, which I realize it could do only if it had access to mapping and routing software, why did it give me any distance at all? Why not just say that it did not know the distance?

I looked into it and found that the distance between any two points on Earth can be calculated quickly using only the global coordinates (latitude and longitude) of the points. This shortest distance is called "great circle distance" because it is measured along a circle, centered at center of the Earth, which is the largest possible circle that encompasses the two given points. If we assume the Earth is a sphere (which it almost is), we can use the Haversine Formula as follows:

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * sin^2\left(\frac{\Delta\lambda}{2}\right)$$
$$angle\ of\ arc\ c = 2 * atan2\left(\sqrt{a}, \sqrt{1-a}\right)$$
$$distance\ along\ arc = R * c$$
$$where\ \varphi\ is\ latitude, \lambda\ is\ longitude, R\ is\ radius\ of\ the\ sphere\ (Earth's\ mean\ radius\ is\ 6{,}371\ km)$$

For this assignment, you are asked to write a C++ program that requests two numbers from the user, a latitude and a longitude and then outputs the shortest distance between the two points. The latitude and longitude can be simple floating-point numbers (using negative for South and West). Feel free to embellish the capabilities if you feel the need, but this first assignment is really just to get you going. You are, however, expected to start using good programming techniques like modularity (functions), appropriate variable names, good comments, etc.

Expected Sample Output

The console window *could* look like the following:

```
24-780B Engineering Computation Prob Set 1 – Distance Calculator

Enter latitude  of point 1 > 40.4433

Enter longitude of point 1 > -79.9512

Enter latitude  of point 2 > 40.7651

Enter longitude of point 2 > -73.9801


Distance is >>> 505.261 km


Press the ENTER key to close the program >
```

Deliverables

1 file, very appropriately named:

ps01_greatcircle_yourAndrewID.cpp

Upload the file to the class Canvas page before the deadline (Tuesday, Sept.7, 11:59pm).

Hint: Even if you name your file appropriately, be sure to include your full name within the code itself (perhaps as a comment block at the top of the file). It is also appropriate to include date and course info, plus a short description of what the program does. (Think about what part of the text in the assignment write-up can be copy/pasted or adapted for this purpose.)

Learning Objectives

Installing and using an integrated developing environment (IDE) and/or editor and compiler (nearly insane choice) for programming in C/C++.

Familiarizing yourself with C++ syntax and how to write a simple program.

Begin to understand how programming style can enhance any program.

Searching references (online and/or textbook) for C++ library functions.