Hack 11.0

Computer Science I
Encapsulation

Department of Computer Science & Engineering
University of Nebraska–Lincoln

Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. You may complete the hack on your own, but you are *highly encouraged* to work with another student and form a hack pair. Groups larger than 2 are not allowed. However, you may discuss the problems *at a high level* with other students or groups. You may not share code directly.

If you choose to form a Hack Pair, you *must*:

1. Both join a hack pair on Canvas (go to People then Hack Pairs)
2. You must both work on the hack equally; it must be an equal effort by both partners. Do not undermine your partner's learning opportunity and do not undermine your own by allowing one parter to do all the work.
3. Turn in only one copy of the code under the individual whose last name comes first (with respect to Canvas).

You are graded based on style, documentation, design and correctness. For detail, see the general course rubric.

Category	Point Value
Style	2
Documentation	2
Design	5
Correctness	16
Total	**25**

Table 1: Rubric

Correctness:

- 5 points for student tester (it is OK to test a function transitively; ie if createAirport uses initAirport and they only test createAirport; both are getting tested)
- 5 points for our "visual" inspection test
- 3 points for the 100 test cases for getAirDistance
- 3 points for the 100 test cases getEstimatedTravel time

Problem Statement

There are thousands of commercial, military, and local airports in the US and around the world. The International Civil Aviation Organization maintains a database of current and inactive airports around the world. The database uniquely identifies each airport by an alphanumeric GPS code. Further, each record contains the following pieces of data on each airport:

- The name of the airport
- Its latitude in degrees in the range $[-90, 90]$ with negative values corresponding to the southern hemisphere
- Its longitude in degrees in the range $[-180, 180]$ with negative values corresponding to the western hemisphere
- The type of airport
- Its elevation in (whole) feet above sea level
- Its municipality and its country

You will design a C structure to encapsulate these attributes to model an airport record from the ICAO database. You will also design several functions to support your structure including factory functions, functions to create a string representation, print records, etc. You will also implement several utility functions that use your structure to compute the air distance(s) between airport locations using their latitude and longitude. Recall that the air distance d between two latitude/longitude points can be estimated using the Spherical Law of Cosines.

$$d = \arccos\left(\sin(\varphi_1) \cdot \sin(\varphi_2) + \cos(\varphi_1)\cos(\varphi_2)\cos(\Delta)\right) \cdot R$$

where

- φ_1 is the latitude of location A, φ_2 is the latitude of location B
- Δ is the difference between location B's longitude and location A's longitude
- R is the (average) radius of the earth, 6,371 kilometers

This formula assumes that latitude and longitude are in radians r, $-\pi \leq r \leq \pi$. To

convert from degrees d ($-180 \leq d \leq 180$) to radians r, you can use the simple formula:

$$r = \frac{d}{180}\pi$$

More details have been provided in a header file, `airport.h`. You will need to design your structure and implement all of the specified functions.

Instructions

- Place all of your function definitions in a source file named `airport.c` and hand it in with your header file, `airport.h`. You may add any utility functions you wish but you must *not* change any of the signatures of the required functions.
- In addition, you must create a main test driver program that demonstrates at least 3 cases per function. Name this file `airportTester.c` and hand it in.
- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.
- You may (in fact are encouraged) to define any additional "helper" functions that may help you.
- Include the name(s) of everyone who worked together on this activity in your source file's header.
- Turn in all of your files via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.