HW 7 – Are We There Yet?

Due Date: Midnight, Tuesday, Nov. 27th

Problem: Given a list of world cities with their latitudes and longitudes, find the cities closest to a user-selected city.

Your assignment: Design, develop and test an Object-Oriented C++ program that will allow the user to select a city from a list, and then will list the five cities (from the same list) that are closest to the selected city, based on Great Circle calculations.

Discussion: The shortest distance between two points on the surface of a sphere is called the Great Circle Distance. Given the latitude and longitude of the two points, that distance can be calculated by use of a publically-available formula. Consider the following ideas in preparing your solution to this problem.

- Develop a user-defined class named City that stores and manages information about a single city, including its name, country, and location. Provide constructor(s) and appropriate methods to manage this information.
- Read and save the contents of the worldcities.csv file into a collection of City instances.
- Repeat the following until the user enters "Quit".
    - Ask the user to pick a city, by name or by index, from all available cities. The user must be able to enter a city number or a city name – both options must be available.
    - Calculate the Great Circle Distance from the selected city to each of the other cities.
    - Find the five closest cities to the one selected, based on the calculated distances.
    - Display the list of five closest cities, along with the great circle distance to each.

Coding

- Each class must be defined within its own set of .h and .cpp files.
- All user input must be case-insensitive and allow embedded blanks. So the user can enter "hong konG" and get a correct match to that city.
- Validate all inputs and do not proceed until valid values are entered.
- Format your source code according to the style guide presented in class.

Bonus:

Extend the program to also display the five cities that are farthest from the selected city.

Reference: http://www.movable-type.co.uk/scripts/latlong.html

Data file: worldcities.csv

Turn in a single zip file containing only your source code in one or more ".h" and ".cpp" files. Name the zip file "First Last HW7", where "First Last" is replaced with your First and Last names.

Rubric:

Issue	Poss.	Earned
TURN-IN		
Source code submitted as a zip file containing only .h and .cpp files	3	3
Source code zip file named correctly	3	3
Source code zip file submitted via Canvas	3	3
Source code zip file turned in on or before the due date	3	3
BASIC FUNCTIONALITY		
Program loads all data from the data file into a collection of City objects (includes parsing the csv file only one time)	20	20
Program presents a menu of appropriate options to the user	8	8
Program validates the user's menu selection before responding to it	5	5
Program allows the user to select a city by entering its index number	15	15
Program allows the user to select a city by entering its name	15	15
Program calculates the Great Circle Distance from the selected city to each of the other cities in the collection.	25	25
Program displays the five closest cities to the selected city, with distances in kilometers	15	15
Program repeats till the user opts to quit	5	5
BONUS FUNCTIONALITY		
Program also displays the five farthest cities from the selected city, with distances in kilometers	15	0
DEBITS – COMPILE AND RUN		
Program compiles with warnings	-10	0
Program compiles with errors	-10	0
Program crashes or runs with errors	-10	0
DEBITS – CODING TECHNIQUES		
No user-defined City class	-20	0
City class has no local instance variables	-5	0
Program does not calculate the Great Circle Distance correctly	-5	0
Methods of the City class do not operate on local instance variables	-5	0
Compares floating point values for exact equality	-5	0
DEBITS - CODING STYLE AND DOCUMENTATION		
Use of one or more goto statements	-10	0
Use of C-strings instead of string variables	-10	0
Call to main()to make the program repeat	-10	0
If statement without a matching else	-5	0
Switch statement not exhaustive and without default case	-5	0
Multiple returns from functions and methods	-5	0
Use of magic numbers	-5	0
Not using descriptive names for variables, constants, enums	-5	0
Not using descriptive names for functions, methods	-5	0
Identifiers not formatted correctly	-5	0

Improper indentation (body of every if, switch, do, while, for, try/catch)	-5	0
No comments in main()	-5	0
No comments in user-defined functions	-5	0
No comments in user-defined class, struct, or enum	-5	0
Total	**120**	**120**