Assignment 1: Civil Engineering Case Study – Aviation Safety

Available: September 24, 2022
Due: 11:55pm October 9, 2022 (Submit via the online submission system on Brightspace)

Programming aspects to get familiarized with:

- Input/Output,
- Selection,
- Repetition.
- File Input/Output

Problem Statement:

Aviation safety is a subject of the highest priority, with aircraft position of paramount importance. Consider a single plane. Two different sensors keep a continuous record of the aircraft location every quarter second. Assume that location is recorded in (*x, y, z*) coordinates, measured relative to the air traffic control tower. The *x*-coordinate is the east-west location (east positive), the *y*-coordinate is the north-south location (north positive), and the *z*-coordinate is the altitude. All values are in kilometres. The first sensor takes its data from a GPS (global positioning system) and stores it in a data file saved to the aircraft "black box". Refer to this file as *file1*. The second sensor takes its data from the air traffic control RADAR system and saves it to another file in the GANS database (In Abu Dhabi, air traffic control is provided by the Global Air Navigation Services (GANS), a private corporation). Refer to this file as *file2*. These two files are later compared by a program to determine data consistency, flagging all times where there is a difference in location greater than some user-defined value.

Assume that the data consist of sequential floating-point values, beginning with the *x*-coordinate, then the *y*-coordinate and then the *z*-coordinate, repeating for an unknown number of time intervals. To compare the two files, the program must prompt the user for a tolerance value, *tolerance*, a floating-point value. Note that *tolerance* is actually a measure of the permissible "distance" between the two recorded locations. The expression for calculating the distance between two locations is:

$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

The program reads the coordinates of a location from each of the two files. It checks whether the calculated distance between the two locations is greater than the specified tolerance. Whenever the difference between the two does exceed the tolerance, the program must record, in the *ERR.LOG file*, the time at which the tolerance was exceeded, and the difference in location (i.e., the "distance" between the data values), together on one line.

Note that the two files may not be the same size. Your solution should accommodate this possibility.

The program should prompt the user about file1 and file2 names to complete the comparison.

Develop a software solution, which will do the following:

- Present the user with a menu for the different functions the program offers, namely (1) calculating the error between the two sensors recordings and saving in the error file, (2) display the error file information, or (3) quit the program.
- In case the user selects the first option (calculating the error and saving into the error file), the program prompts the user for the respective inputs (tolerance value and the two data file names), validates the user input whenever reasonable, calculates the errors, and save the error values that exceed the tolerance in the *ERR.LOG* file. The program then returns to the main menu.
- If the user selects the second option, the program displays the content of the *ERR.LOG file* in a tabular format. An error message must be printed in case the *ERR.LOG* file does not exist. The program then returns to the main menu.
- If the user selects the exit option, the program prints a terminating message and quits the program.
- Make sure your code is well commented.
- You can make additional assumption in case you feel any information is missing. Make sure to clearly state them.

Submission

Your submission **must** include an assignment report (Word Document or PDF) and a C++ source code (.cpp file). The report must follow the five steps model for software development (as discussed in class):

a) Step 1: Problem Identification and Statement (5 points)
b) Step 2: Gathering Information (10 points)
c) Step 3: Test Cases and algorithm (35 points)
d) Step 4: Code or implementation (35 points)
e) Step 5: Test and Verification (a minimum of 4 test cases) (15 points)