Assignment 1: Civil Engineering Case Study – Freezing Temperature of Seawater

Available: Monday 2 October 2023
Due: Sunday 15 October 2023 @11:55pm (Submit via Brightspace)

Programming aspects to get familiarized with:
- Input/Output,
- Selection,
- Repetition.
- File Input/Output

Problem Statement:

Seawater is primarily water with about 3.5% dissolved material (salts, metals, and gases) from volcanic eruptions and the weathering of rocks. The saltiest ocean waters are in the Atlantic around the equator, where evaporation exceeds precipitation. The salinity of seawater is a measure of the amount of dissolved material in the seawater. Salinity is often measured using an instrument that measures the electrical conductivity of the water; the more dissolved materials in the water, the better it conducts electricity. Measurements of salinity are especially important in colder regions because the temperature at which seawater freezes depends upon its salinity.

Assume that a file stores a set of salinity and corresponding freezing temperature measurements (the total number of measurements in the file is unknown to the user), develop a software that:

1. Provide the user with a menu of three options: (1) calculate the linear regression model using the data stored in the file, (2) calculate a freezing temperature given the water salinity using the linear model developed in option (1), and (3) exit the program.
2. For option (1), the program prompts the user to enter the file name including the file extension. The program then reads the data from the file and determines the linear interpolation function for the relationship between the freezing temperature and the salinity using the following model:

$$T = m \times S + b \text{ , where} \quad (1)$$

T is the freezing temperature, *S* is the water salinity, and *m* and *b* are the model parameters. The model parameters *m* and *b* can be calculated as follows (considering n number of data entries that are determined based on the file content):

$$m = \frac{\sum_{k=1}^{n} x_k \cdot \sum_{k=1}^{n} y_k - n.\sum_{k=1}^{n} x_k y_k}{(\sum_{k=1}^{n} x_k)^2 - n \cdot \sum_{k=1}^{n} {x^2}_k}$$

$$b = \frac{\sum_{k=1}^{n} x_k \cdot \sum_{k=1}^{n} x_k y_k - \sum_{k=1}^{n} {x^2}_k \cdot \sum_{k=1}^{n} y_k}{(\sum_{k=1}^{n} x_k)^2 - n \cdot \sum_{k=1}^{n} {x^2}_k}$$

As an example, if the file stores the following data (first line stores the table header):

Salinity (ppt)	Freezing Temperature (F)
0	32
10	31.1
20	30.1
24.7	29.6
30	29.1
35	28.6

Then to evaluate these equations, we need to compute the following group of sums:

$\sum_{k=1}^{n} x_k = 0 + 10 + 20 + 24.7 + 30 + 35 = 119.7$

$\sum_{k=1}^{n} y_k = 32 + 31.1 + 30.1 + 29.6 + 29.1 + 28.6 = 180.5$

$\sum_{k=1}^{n} x^2{}_k = 0^2 + 10^2 + 20^2 + 24.7^2 + 30^2 + 35^2 = 3235.09$

$\sum_{k=1}^{n} x_k y_k = 0 * 32 + 10 * 31.1 + 20 * 30.1 + 24.7 * 29.6 + 30 * 29.1 + 35 * 28.6 = 3518.12$

Under these sums, we can now compute the values of m and b:

m = - 0.0978
b = 32.0347

3. Option (2) assumes that a linear model is already calculated. The program prompts the user for a salinity value (should be larger or equal to zero) and calculates the freezing temperature using the model equation (1). The program displays the freezing temperature on the output screen.
4. If the user selects the exit option, the program prints a terminating message and quits the program.
5. The program must handle invalid menu option provided by the user. For example, if the user enters -1, the program should display a message reporting invalid choice and present the menu again.
6. Make sure your code is well commented.
7. 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)