CPT106 C++ Programming and Software Engineering II

Assessment 3

Assessment Number	3
Contribution to Overall Marks	35%
Release Date	Tuesday, 20-April-2021, 23:59
Submission Deadline	Wednesday, 5-May-2021, 23:59

How the work should be submitted?

SOFT COPY ONLY !

(This is an INDIVIDUAL WORK and MUST be submitted through Learning Mall so that the TAs can run your programs during marking.)

1. Make sure your name and ID are printed on the cover page of your report.
2. **A short report** named with your **student ID number** with up to a few pages in **PDF format** should be submitted to Learning Mall**.**
3. **C++ source code files** for all questions should be zipped into ***a single file named with your student ID number*** and submitted to Learning Mall.
4. The report and zipped source code file should **be submitted separately** to two different submission boxes on Learning Mall.
5. Late Submission Policy: 5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of five working days.

Assessment Overview

This assessment aims at testing some basic concepts of C++ programming and initiates the routine of code development using the software development process (**SDP**). The report should include five main steps (step 1-5 below) of the software development process, and the marking criteria is as follows:

1. Problem statement: formulate the problem. (10 marks)
2. Analysis: determine the inputs, outputs, variables, classes, class relationships, etc. (10 marks)
3. Design: use flow chart or UML activity diagram to define the main steps (the algorithm) needed to solve the problem. (10 marks)
4. Implementation: the C++ code has to be submitted as a zipped file. Just indicate here the name of the file.(40 marks, further divided in page 3)
5. Testing: explain how you have tested and verified your C++ program and include some sample runs of your C++ Programs. **Testing result must be shown by screenshot and included in your report.** (20 marks)
6. The overall quality of report, including structure, writing, etc. (10 marks)

Line fitting algorithms

Given a set of x,y data points, it is often necessary to automatically calculate an equation which gives the best fit line through the data. This type of analysis is known as line fitting or regression analysis.

The mathematical basis behind line fitting is quite straightforward. You have a set of data points represented by (x1,y1), (x2,y2) ..... (xn, yn). You want an equation which represents this data, the exact type of equation will depend on the way in which the data varies. Given any mathematical relationship between x and y it should be possible to generate a line fitting algorithm. The line fitting algorithm is described below.

Straight line

The equation for a straight line is: y = ax + b. Suppose that we have n points of (x,y) values from the text file, the values of ***a*** and ***b*** for **the best fit line** that is the best approximation of the given set of data points can be obtained from the following equation:

$$a = \frac{\begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \quad , b = -\frac{\begin{vmatrix} a_1 & d_1 \\ a_2 & d_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

where

$$a_1 = \sum_{i=1}^{n} (x_i * x_i)\,, \qquad a_2 = \sum_{i=1}^{n} x_i\,,$$

$$b_1 = \sum_{i=1}^{n} x_i\,, \qquad \mathrm{b}_2 = \mathrm{n}\,,$$

$$d_1 = -\sum_{i=1}^{n} (x_i * y_i)\,, \qquad d_2 = -\sum_{i=1}^{n} y_i$$

The value of a second order determinant $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ is the difference between the two dot products (a_1b_2 - a_2b_1). It is easy to create a C++ program to calculate its value.

Distance

The distance from a point ***P(x,y)*** to the line ax+by+c=0 is the length of the perpendicular line from the point ***P*** to the line:

Distance = $\frac{|\mathrm{ax+by+c}|}{\sqrt{a^2+b^2}}$
where (x,y) is the coordinate value for point ***P***.

Standard error of distance for all the points: SE=$\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}}$

Where $\bar{x}$ is the mean distance of all points, x_i is the distance of Point P_i , n is the number of all points.

Please complete the assessment according to the following steps: (40 marks)

1. Create a file operation class “FileOp” with functions as follows: (12 marks, 3 marks per question)
   a) It can allow users to input points from the keyboard and save these points into a text file.
   b) It can read points from the text file.
   c) It can delete a point from the text file.
   d) It can check if a point is in the text file.
2. Create a class “LineFitting” with functions as follows: (15 marks, 3 marks per question)
   a) It can calculate the parameters of a, b and c for the best fit line ax+by+c=0.
   b) It can calculate the distance between each point in your file and the best fit line.
   c) It can find the best point from the file, which is not in the best fit line but has the minimal distance compared to other points.
   d) It can find the worst point from the file, which has the maximal distance compared to other points.
   e) It can calculate the standard error of distance for all the points.
3. Create a main method in you program to achieve follow functions: (13 marks)
   a) Create a main method. (1 mark)
   b) Create a text file using the “FileOp” class with some points. (2 marks)
   c) Print out the equation of the best fit line (e.g. 3x+2y+3=0). (2 marks)
   d) Ask the user to input a point and print out whether the point is in the text file or not, if it is in the file then print out its position. (2 marks)
   e) Print out the worst point that has a maximal distance to the best fit line in all the points. (2 marks)
   f) Print out the standard error for all the points. (2 marks)
   g) Ask user to input a point and delete it from the file if the point exists in the text file. (2 marks)