CS 200 – Intro to Programming

Assignment 1 Fall 2018

Due Date: Monday 8th October 2018, 11:55 PM

Please keep in mind the following guidelines:

- *Do not share your program code with anyone.*
- *Do not copy code from the internet.*
- *If you receive any assistance, mention the part of code in*

which you received assistance.

- *You must be able to explain any part of your submitted code.*
- *All submissions are subject to automated plagiarism detection.*

Submission:

You have to submit all the .cpp files containing source code. Zip all .cpp files into one file named as <your8DigitRollNumber>_Assignment1_CS200.zip and submit the zip file.

Total Marks: 100

Question 1: (15 marks)

Note: Both parts are different of each other so you are required to implement two separate files.

Part a: (5 marks)

Suppose you are given an array with integer values. You have to write a function to generate a frequency distribution from this array. For example, if the array has:

1 1 1 2 1 2 3 1 1 3 2 5 9
then as output the function should prints the two arrays, one with the values and the other with the frequencies. Make sure the values array does not have duplicate entries.

ValuesArray: {1, 2, 3, 5, 9}
FrequencyArray: {6, 3, 2, 1, 1}

Hence the prototype of this function should look something like this:
void FindFrequencyDistribution(int valuesArray[], int FrequencyArray[], int inputArraySize, int FrequencyArraySize).

Part b: (10 marks)

You have to implement a matrix multiplier. Following things must be taken into account.

- ➔ User is asked to input the rows and columns of matrix one and two.
- ➔ Then user is asked to input the elements of the matrices.
- ➔ Write a function that prints addition of both matrices.
- ➔ Write a function that prints subtraction of both matrices.
- ➔ Write a function that returns the multiplication of both matrices and print the returned matrix in main function.
- ➔ Write a function that returns the inverse of both matrices and print the results in main function.

Question 2: (25 marks)

Part a: (10 marks)

In this part you will be given a file containing a few pages of an article. Your job is to implement a class Word_Index that will take the article in its constructor and store all the words in an array. Be careful about repeated words; a word should only appear once in your array.

Part b: (10 marks)

Create a function that prints out each individual word in the passage and the corresponding line numbers it appears in. Your class now has two private variables: One is the array of words and the other is an array of the lines the word appears in. If the word appears twice in a single line, include the line number twice as well. Remember that a line break is denoted by '\n'.

For example, with the following passage as input:

"I like apples. Apples

grow on trees. Trees are

green."

Your output should be:

Apples: 0, 0

Are: 1

I: 0

Like: 0

On: 1

Green: 2

Grow: 1

Trees: 1, 1

Part c: (5 marks)

In this part you are to implement a function that will take your array of words as an argument and return a sorted version of the array in alphabetical order.

Note: This question involves usage of file handling, arrays and random functions.

Question 3: (60 marks)

Part a: (20 marks)

You are asked to save Roll Number, Marks and Grade of 100 students in "Students.txt" file. Following things should be taken into account:

- ➔ You have to write this data line-by-line in file "**Students.txt**". If file is not created yet, create a new file and open the file in append mode. If File is already present, clear all the previous contents of the file first. Be sure to close the file after the working is complete.
- ➔ Generate roll number randomly. Range should be from 0 to 100. There should be no repetitions.
- ➔ Marks should be generated randomly. Range should be from 20 to 95.
- ➔ Write a function "**Grade_Calculation()**" in which a student will get an 'A+' grade if his/her marks are greater than 89. Otherwise his grade calculation will be as follows
  - o If marks are greater than 79 than Grade equals A
  - o Else if marks are greater than 69 than grade equals B
  - o Else if marks are greater than 59 than Grade equals C
  - o Else If marks are greater than 49 than Grade equals D
  - o Else Grade equals F.
- ➔ Save all the data into the file line-by-line in the format "roll_number,marks,grade".

Part b: (15 marks)

In this part you are required to make a header file "**Student.h**" in which you are required to make a class Student. Following things should be taken into account:

- ➔ The class should have private variable "**roll_number**", "**marks**", "**grade**". It should have a default constructor that initializes all variables with appropriate values. You have to decide what appropriate variable types must be defined.
- ➔ It should have overloaded set function, that either takes roll number and marks together or grade.
- ➔ It should have 3 get functions, one for each variable.
- ➔ It should have a friend function named "print" that takes object of Student class as input and print all the values in the format

  Student <roll number> has <marks> Marks and got <grade> Grade.

  Note that roll number, marks and grade should be fetched using get functions of the class.

Part c: (25 marks)

Read the file "Students.txt". Following things should be taken into account:

- ➔ Make an array of 100 "**Student**" class objects by importing the "**Student.h**" file.

- ➔ Read the file line-by-line. Save the input into the string. Separate the string upon “comma”. Type cast the roll number and marks into appropriate data type and save all the data into objects created earlier. Note that one object can have data of only one student.
- ➔ Create a function “**Desired_Student()**” that takes the roll number and object array as input and prints the desired student. The roll number should be passed by reference and array should be passed by value.
- ➔ Write a function “**HighestScorer( )**” which takes Object array as input and outputs the student details as output.
- ➔ Write a function **AverageResult( )** which outputs mean(average) of class results.
- ➔ This all should be called in main function. Note that user must be asked which function he/she wants to run, does she want to create a new file filled with student details or use the previous one, and does he/she wants to exit the program.

Bonus:

- ➔ Write a function “**Calculations()**” that takes Students information and outputs the standard deviation, median, mod ,variance of the marks without using built-in functions.
- ➔ Follow this link for further help :
  - https://www.bmj.com/about-bmj/resources-readers/publications/statistics-square-one/2-mean-and-standard-deviation
  - https://sciencing.com/median-mode-range-standard-deviation-4599485.html
- ➔ Write a function “**Sorter()**” that asks the user that by which means the user wants to sort the Students (i.e by student roll number, marks or grade) and write the sorted data into a new file “**Sorted_Students.txt**”.