**Challenge 4**

**Instructions**

1. You can write the code in java language.

2. Upload the code to github with the support of your mentors.

3. Share the link of your github repository on or

**Problem**

Read student information from a file, store the data, compute final grades, and then print a summary report to an output file.

**Program Details**

1. Design a set of classes that store student grade information. There should be one base class to store common data and three derived classes that divide the set of students into three categories: English students, Science students, and Math students. All data stored in these classes should be private or protected. Any access to class data from outside should be done through public member functions. The base class should allocate storage for the following data (and only this data):
   1. student’s first name (assume 30 characters or less)
   2. student’s last name (assume 25 characters or less)
   3. Which course the student is in (English, Science, or Math)
2. Each class should have a function that will compute and return the student’s final average based on the stored grades. All grades are based on a 100-point scale. Here are the grades that need storing for each subject along with the breakdown for computing the final grade: ◦

**English**

* + 1. Term Paper = 30%
    2. Midterm = 30%
    3. Final Exam = 40%

**Science**

* + 1. Attendance = 10%
    2. Project = 30%
    3. Midterm = 30%
    4. Final Exam = 30%

**Math**

Quiz Average = 15%

There are a total of 5 quizzes averaged together (result can be a decimal number)

1. Test 1 = 30 %
2. Test 2 = 20 %
3. Final Exam = 35%

**File Formats**

**Input File**

The first line of the input file will contain the number of students listed in the file. This will test you how big a list you need. After the first lines, every set of two lines constitutes a student entry. The first line of a student entry is the name in the format lastName, firstName. Note that a name could include spaces; that the comma will delineate last name from first name. The second line will contain the subject (“English”, “Science”, or “Math”) followed by a list of grades (all integers) all separated by spaces. The order of the grades for each class type is as follows:

English – Term paper, Midterm, Final Exam

Science – Attendance, project, Midterm, Final Exam

Math – Quiz 1, Quiz 2, Quiz 3, Quiz 4, Quiz 5, Test 1, Test 2, Final Exam

**Output File**

The output file that you print should list each student’s name (firstName lastName – no extra punctuation between), Final Exam grade, final average (printed to 2 decimal places), and letter grade based on a 10-point scale (i.e., 90-100 A, 80-89 B, etc.). Output should be separated by subject, with an appropriate heading before each section, and each student’s information listed on a separate line in an organized fashion. Data must line up appropriately in columns when multiple lines are printed in the file. At the bottom of the output file, print a grade distribution (i.e., how many As, Bs, Cs, etc.) of all students.

**General Requirements**

No global variables other than constants

All class member data must be private or protected

Use the const qualifier on member functions wherever it is appropriate

**Evaluation Criteria**

**Logic related**

1. Understanding **object orientation** andimplementation of classes including appropriate access modifiers. **10 marks**
2. Correct implementation of the logic to manage file read, data processing and file write operations. **10 marks**
3. Capability to clearly explain the implementation. (Take a discussion with the mentee to mark) **10 marks**

**Code related**

1. Correctness of defining variables, data structures **2 marks**
2. Readability of the code (Cognitive complexity) **2 marks**
3. Empty check and/or null check if applicable **1 marks**
4. Executes without run time errors **5 marks**

Total **30 marks**