



GRIFFITH COLLEGE DUBLIN

COMPUTING ASSIGNMENT TITLE SHEET

Course:	B.Sc. in Computing (Level 7 and Level 8)
Stage/Year:	I
Module:	Computer Programming
Semester:	II
Assignment Number:	V
Date of Title Issue:	7/04/20
Assignment Deadline:	28/04/20
Assignment Submission:	Submitted on Moodle
Assignment Weighting:	10%

Assignment Title

Project Marking Assistant

Make an application that opens a .csv file that contains project marks for the whole class and lets the lecturer display and edit marks.

Store everything in objects. The project should have 2 states marked and unmarked.

When the application is launched, it should first ask the user to enter a file name.

User has to type file name with extension in presumable directory of there.

If the file doesn't exist, the program should exit. Otherwise it should read the file.

The file is expected to have the following structure. The first row should contain column names. The first two columns should always have the same name: student id and student name. The last column name should also always be the same name: total. The rest of the column names are various marking criteria. There can be **any number** of these columns, they must always have a number of assigned points in brackets to the **right** of the column name and the sum of these numbers must be 100. If the sum is not 100 or some of the numbers are missing, the user should be asked to enter the marks for each of the columns one after another. This process should repeat until the sum is 100. An example of the first row of a valid csv file is given below: **1st The program knows the purpose of the first and last column.**

Program should be able to handle any number of columns.

(Example our file doesn't have to be the same it could have completely different criteria 6 marking criteria min)

```
Student Id, Student Name, Technical Difficulty (30),
Innovation (20), Documentation (25), Presentation (25), Total
30+20+25+25=100
```

Note that your csv file should have at least 6 marking criteria that are different from the example and your program should be able to handle files with different numbers of marking criteria.

(15 marks)

The program should then read one row after another and store details from each row in a separate instance of a Project class. You will need to design the Project class yourself. The program should check that the total is correct and if it is not correct, it should be replaced with the correct value. **2nd Use classes to store multiple types of data using classes as one element of an array can reference a class for multiple data information stored. (The Class should store ID (String) Name (Strings), Don't use Int but Double and round up to nearest 2 digits to get a precise mark for storing number data.**

(15 marks)

Your program should have a method that displays the current state of the entire marking table. It should first display the column names (together with numbers in brackets) and then details of each project. The data should be arranged in such a way that data aligns with the corresponding column name. You may need to use the Tab character. If a mark is missing, a dash (minus sign) should be displayed instead.

(15 marks)

After the program displays the marking table it should show a menu with options. Every time a task is completed the updated marking table and the menu should be displayed. The menu should have the following entries:

1. Enter project marks.
2. Add new project.
3. Delete project.
4. Save and Exit.
5. Exit without saving.

(10 marks)

(Let the lecturer remark project and overwrite ID data entered.). Use a student id check method as it will be required multiple times.

If the user selects "Enter project marks" the user should be asked to enter Student Id. **If the entered Student Id cannot be found**, the user should be alerted about this and return to the menu. Otherwise the user should be asked to enter one mark at a time. Each time the user should be informed what the marking criteria is called and what maximum mark can be entered. For example: "Technical Difficulty 0-30". If the entered mark is less than zero or greater than the maximum mark the user should be asked to enter it again until they enter a valid mark. After all marks are entered, the total is calculated and displayed on the screen.

(15 marks)

If the user selects "Add new project" they will be asked to enter Student Id and Name. The program should check that this **Student Id doesn't yet exist**, and if it does, ask the user to enter it again. The program should also check that Student Id follows the following format: **it must start either with P or F (use ASCII to count if it's one other other within a range)** followed by 6 or 7 digits. Then the user should be asked whether they want to mark the project as well. If they do, the marking process should be the same as described above.

(15 marks)

If the user selects "Delete project", they should be asked to enter Student Id, and if it exists, delete the corresponding project.

QA EA3

(5 marks)

“Save and Exit” means that the file from which we read data should be overwritten with updated data. Note that the resulting file is expected to be a valid correctly formatted .csv file.

(10 marks)