

## Assignment 2

### Input and Output Operations

---

#### Overview

- This assignment is to be conducted **outside of the class**.
- You will be adopting a **Pair Programming** strategy in doing this assignment.  
[What is pair programming?](https://youtu.be/oBraLLybGDA) (<https://youtu.be/oBraLLybGDA>)
- You and your partner will be coding collaboratively online using VS Code and **Live Share**.
- You will communicate with each other using an online meeting software such as Webex, Google Meet, etc.
- You will record the pair programming session. This requirement is compulsory.

#### Pair Programming and Collaborative Coding

- Select a two-hour time slot within the given date to engage in a pair programming session with your partner.
- If necessary, split the pair programming session into multiple sub-sessions, ensuring that the total time does not exceed 2 hours.
- Document the date and time of each pair programming session in the program's source code.
- Keep a record of the meetings pertaining to your pair programming sessions. This includes documenting discussions and decisions made during the sessions. If the programming occurs over multiple sessions, ensure all meetings are recorded. No need to edit the video footage.
- While face-to-face pair programming is an option, remember to still record the session to maintain transparency and accountability.

#### Notes:

- Before engaging in the pair programming session with your partner, it is recommended that you individually explore the exercise first. This proactive approach will better prepare you for the collaborative coding experience and enhance the overall effectiveness of the session.

## **Recording the Pair Programming Sessions**

- Utilize any online meeting tool such as Webex, Google Meet, etc. for conducting online meetings and recording your pair programming sessions.
- Note that (if you are opting to use Webex), the free account on Webex limits meetings to 50 minutes per session. If more time is needed, open another session once the current one ends.
- Since the free account on Webex only allows local recording, ensure to record the session on your computer. Later, manually upload the videos to a cloud storage platform such as Google Drive.

### **Purpose of the Video:**

- Emphasize that the video is intended for documenting the pair programming session, rather than for presentation purposes.
- It should capture the coding process, communication, and collaboration between you and your partner.
- Use English for communication.

### **Video Content:**

- Display your Visual Studio Code (VS Code) interface and the output (console terminal) during the coding session.
- Ensure that your camera is turned on throughout the session to show both participants.
- You may record the session in a single video or multiple segments.
- Submit the raw, unedited videos without any post-processing.

### **Uploading and Sharing:**

- Upload the recorded videos to your Google Drive or YouTube channel.
- If using Google Drive, organize multiple videos into a single folder and share only the folder link. Set permissions so that "Anyone can view" the videos.
- If uploading to YouTube, provide links to all the videos.
- Ensure that the videos remain accessible until the end of the semester.

## **Plagiarism Policy Notice**

While collaboration and consulting resources are encouraged, it is imperative to uphold academic integrity. Any instance of plagiarism will result in immediate dismissal of your submission. There will be no opportunity for appeal.

## **Late Submission Policy and Penalties**

- All submissions must be made through the designated eLearning platform. Submissions via other channels such as email, Google Drive, or Telegram will not be accepted.
- In case a program fails to compile, a penalty of 50% will be applied to the submission.
- Late submissions will incur penalties as follows: For every hour past the deadline, a penalty of 10% will be deducted. The calculation of late penalties will be rounded up to the nearest hour. For example, a submission that is 1 minute late will be considered 1 hour late.

## Problem

In this assignment, your task is to develop a C++ program that generates performance reports for a class of multiple students. The program should be flexible, allowing it to process different classes by accepting input from a text file. Upon execution, the program should prompt the user to enter the input file and then generate the report as screen output. The following sections explain the details requirements about the program:

### 1. User Input:

The program will request the user to input a file name. It will then verify the existence of the specified input file. If the file cannot be open, the program will prompt the user to retry the input. The example below illustrates the user interaction with the program. User inputs are indicated by bold text.

```
Please enter the file name: pt2.txt

Sorry! Unable to open the file. Please try again

Please enter the file name: pt1.txt
<Afterward, the program proceeds generating the output>
```

### 2. Input Files:

The following figure shows an example of input file named **pt1.txt**

```
1 SECJ1013 Programming Technique I
2 6
3
4 A21CS8024 50.25 30.75
5 A21CS7012 45.75 34.75
6 A21CS6098 55.25 25.75
7 A21CS5003 40.75 19.75
8 A21CS4001 48.25 32.25
9 A21CS3005 42.25 29.75
```

Here is the description of the format of each input file.

Line	Description
1	Course code and Name
2	Total number of students in the class
3	blank line

4 onwards	Each subsequent line contains a student's matriculation number, followed by their coursework and exam marks, respectively.
-----------	--

### 3. Screen Output (Class Performance Report):

The following figure shows an example output produced from the above input file (pt1.txt)

Performance Report for SECJ1013 - Programming Technique I					
No	Student Matric No.	Coursework Mark	Exam Mark	Total Mark	Grade
1	A21CS8024	50.25	30.75	81.00	A
2	A21CS7012	45.75	34.75	80.50	A
3	A21CS6098	55.25	25.75	81.00	A
4	A21CS5003	40.75	19.75	60.50	B-
5	A21CS4001	48.25	32.25	80.50	A
6	A21CS3005	42.25	29.75	72.00	B+
Class Average:		47.08	28.83	75.92	A-

Here is the description about the output:

- The Total Mark is obtained by the sum of the coursework and exam marks
- The grade is obtained based on the total mark following the grading system below:

mark	Grade
90 - 100	A+
80 - 89	A
75 - 79	A-
70 - 74	B+
65 - 69	B
60 - 64	B-
55 - 59	C+
50 - 54	C
45 - 49	C-
40 - 44	D+
35 - 39	D
30 - 34	D-
0 - 29	E

- Each column should be allocated with an appropriate width. You should identify the width accordingly.
- Each column is aligned to the left.
- Each mark should be displayed in 2 decimal points

#### 4. User-defined Functions:

Your program should explicitly define several functions below:

- `getInputFile()` -reads the input file's name. See the details in Requirement 1.
- `grade()` -determines the letter grade for a given marks See the details in Requirement 3.

The following figures show example runs of the program processing different input files. Bold texts indicate user inputs.

##### Example Run 1

Please enter the file name: **pt1.txt**

Performance Report for SECJ1013 - Programming Technique I

-----					
-					
No.	Matric	Coursework	Exam	Total	Grade
-----					
-					
1	A21CS8024	50.25	30.75	81.00	A
2	A21CS7012	45.75	34.75	80.50	A
3	A21CS6098	55.25	25.75	81.00	A
4	A21CS5003	40.75	19.75	60.50	B
5	A21CS4001	48.25	32.25	80.50	A
6	A21CS3005	42.25	29.75	72.00	A-
-----					
-					
Class Average:		47.08	28.83	75.92	A-

##### Example Run 2

Please enter the file name: **oop.txt**

Performance Report for SECJ2153 - Objecct-Oriented Programming

-----					
-					
No.	Matric	Coursework	Exam	Total	Grade
-----					
-					
1	A21CS9001	54.50	25.50	80.00	A
2	A21CS9002	48.75	30.25	79.00	A-
3	A21CS9003	50.25	35.75	86.00	A
4	A21CS9004	45.75	30.25	76.00	A-
5	A21CS9005	56.25	19.75	76.00	A-
6	A21CS9006	49.50	24.50	74.00	A-
7	A21CS9007	52.75	27.25	80.00	A
8	A21CS9008	57.50	22.50	80.00	A
9	A21CS9009	49.25	31.75	81.00	A
10	A21CS9010	35.00	16.00	51.00	C+
-----					

-  
Class Average:      49.95                      26.35                      76.30                      A-

## Assessment

This exercise carries **7%** weightage for the final grade of this course. The breakdown weightage is as follows (out of 100 points):

Criteria	Points
<b>1. The code</b>	
a. Reading data from the input file	15
b. Calculating the total mark and grade for each student	15
c. Printing the output with correct format	20
d. <code>getInputFile()</code>	10
e. <code>grade()</code>	10
<b>2. Pair Programming</b>	
a. Overall	10
b. Active collaboration	10
c. Both members play both roles Driver and Navigator.	10

**Notes:** Pair programming and the video are compulsory. If your source code submission is not accompanied by the video, your assignment will not be assessed..

## Submission

- Deadline: As specified on eLearning
- Only one member from each pair needs to do the submission.
- Submission must be done on eLearning. Any other means such as email, telegram, google drive will not be accepted at all.
- You will need to submit TWO (2) items:
  - a. Source code: submit only the source code file, e.g. `main.cpp`. Write your name and your partner's in the source code.
  - b. The video link of your pair programming session. Write the link in the source code.



## FAQs

### 1. Who will be my partner?

You will choose your partner on your own.

### 2. Can I do the exercise alone?

This is only allowed if the number of students in the class is not even. You also need to ask for permission from the lecturer.

### 3. Do we need to switch roles between Driver and Navigator?

Yes. Your video should show that you and your partner keep switching between these two roles. No one should be dominant or play only one role.

### 4. What if we do pair programming physically (face-to-face)?

You and your partner should use only one computer and sit side-by-side. You do not have to open LiveShare and online meetings. You can record the video locally using software like OBS. Again, you still need to talk and discuss with your partner in the video. It is also compulsory to turn on the web camera.