Exercise 1
Introduction to Classes and Objects

Overview

- This exercise is to be conducted **outside of the class.**
- You will be adopting a **Pair Programming** strategy in doing this exercise.
  What is pair programming? (https://youtu.be/oBraLLybGDA)
- You and your partner will be coding collaboratively online using VS Code and **Live Share**.
- You will communicate to each other using Webex, an online meeting software.
- You will record the pair programming session.

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 within 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 Webex for conducting online meetings and recording your pair programming sessions.
- Note that 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 or Bahasa Malaysia 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

This exercise entails developing a C++ program that determines the grade earned for a given subject score, employing an Object-Oriented Programming (OOP) paradigm.

Notes:

- Write the program in a single source code file and use the inline style to define the class. This means that the definition of each method is written directly in the class declaration.

- Follow proper naming convention:
  - Use camel Case to name functions, methods, and variables.
    - Example: `int calculateTotal(), int thisIsVariable`
  - Use Pascal Case to name class and data type.
    - Example: `class Student{ }`
  - Use CAPITAL case to all characters to name constants.
    - Example: `const int MAXIMUM_STUDENT = 10;`
  - Use small case to all characters to name a file
    - Example: `my_main_program.cpp`

Modify the starter program provided (main.cpp) to accomplish the following tasks:

1. Declare a class to model subject information which consists of three attributes, the subject's name, code, and the score earned for the subject. In your code, you should also implement the "data hiding" principle for the class.

2. Define two constructors for the class as follows:
   a. a **parameterized constructor** which accepts parameters to set the attributes.
   b. a **default constructor** method which sets the attributes to "empty" values

3. Define the **destructor** method for the class which does nothing.

4. Define an **accessor** method (also known as **getter**) and a **mutator** method (also known as **setter**) for **each attribute** of the class. For example, as for the attribute `code`, define two methods for it, i.e., **`getCode()`** and s**`etCode()`**. Keep in mind that the accessors should be declared as **constant methods**.

5. Define three more **accessor methods** below:
   a. a method that determines the grade earned for the subject based on the score. For example, if the score earned is `70`, this method should return `"B+"`.
   b. a method that determines the point value of the grade earned. For example, if the grade earned is `"B+"`, this method should return `3.33`.
   c. a method that determines the point earned for the subject. For example, if the subject is 3 credit hour and the grade point earned is `3.33` , this method should return `9.99`

   See the section below for the formulas.

6. In the main function, write the code to accomplish the following requirements:
   a. Instantiate or create an object from the class.
   b. Read inputs from the user and use them to set the attributes of the object.
   c. Print the subject information such as the code, name, score, grade, etc. See example runs in the following figures for the expected results.

   Example Run 1

```
Enter the following data:
  Subject name => Programming Technique II
  Subject code => SECJ1023
  Score earned => 95


THE RESULT

Subject Code : SECJ1023
Subject Name : Programming Technique II
Credit Hour  : 3
Score Earned : 95
Grade Earned : A+
Grade Point  : 4
Point Earned : 12
```

Example Run 2

```
Enter the following data:
  Subject name => Graduate Success Attributes
  Subject code => UHMT1012
  Score earned => 84


THE RESULT

Subject Code : UHMT1012
Subject Name : Graduate Success Attributes
Credit Hour  : 2
Score Earned : 84
Grade Earned : A
Grade Point  : 4
Point Earned : 8
```

Grade and Point Calculation and Formula

The grade and point value earned are determined based on the following table.

Score	Grade	Point Value
90 - 100	A+	4.00
80 - 89	A	4.00
75 - 79	A-	3.67
70 - 74	B+	3.33
65 - 69	B	3.00
60 - 64	B-	2.67
55 - 59	C+	2.33
50 - 54	C	2.00
45 - 49	C-	1.67
40 - 44	D+	1.33
35 - 39	D	1.00
30 - 34	D-	0.67
0 - 29	E	0.00

Credit hour is determined from the last digit of the course code. For example, the course with code SECJ1023 is 3 credit hours

Point Earned = Point value x credit

Assessment

This exercise carries **2.5%** weightage for the final grade of this course. The breakdown weightage is as follows (out of 100 points):

Criteria	Points
1. The code	
a. Task 1 – class declaration and class attributes	10
b. Task 2 – constructors and destructor	10
c. Task 3 – mutators and accessors (for the attributes)	20
d. Task 4 – three more getters	15
e. Task 5 – main function	15
2. Pair Programming Session	
a. Overall	10
b. Active collaboration	10
c. Both members play both roles Driver and Navigator.	10

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.
- Submit only the source code (i.e., main.cpp)

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.