School Computing, Faculty of Engineering
Universiti Teknologi Malaysia

Programming Technique II
Semester 1, 2024/2025

Exercise 2
Class and Object Manipulations

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 or other tools that fit the same purpose.
- You will communicate to each other using Webex or other online meeting softwares.
- You will record the pair programming session.

Pair Programming and Collaborative Coding

- Pick any time worth **TWO (2) hours** (maximum) within the given date to conduct the pair programing session with your partner.
- You may also split your pair programming into several sub-sessions provided the total time is still within 2 hours.
- Log the date and time for every pair programming session conducted. Write them in the program source code.
- Record the meeting about your pair programming session. If you do your programming in multiple sessions, record all of them. You do not have to edit the video.
- Code submissions without the video at all or the video was too short, will be declined.

Notes:

- You are advised to explore the exercise on your own first before doing the pair programming session with your partner. This should make yourself be more prepared.

How To Record the Session

- Use your preferred online meeting software to conduct the online meeting and to record your pair programming sessions.
- Free account Webex only allows 50 minutes of meeting per session. Thus, should you need more time than that, you will need to open another session once the current one ends.
- Free account Webex only does not allow recording in the cloud, but only for local recording, i.e. the video will be stored on your computer. Thus, later you will need to upload the videos to the cloud (e.g., to Google Drive) manually.

About the Video

- The video is not meant for presentation purposes, but for recording your pair programming session.
- The video must show that you are coding, communicating, and collaborating with your partner. In this regard, speak in English or Bahasa Malaysia.
- In the video you should show your VS Code and the output (console terminal). Also, you need to turn your camera on.
- You can record the session in a single or multiple videos.
- Upload the videos to your google drive or YouTube.
- If you upload multiple videos on Google Drive, put them in a single folder, and submit only the folder's link. Set the video file (or folder) permissions so that **"Anyone can view"**. If you upload the videos on YouTube, submit all the video links.
- Make sure the video is available until the end of the semester.
- Submit the raw videos, i.e., you don't have to do post-editing.

Notes:

- Please make the font-size of your VS Code a little bit larger so that it easy for me to see your code in the video. You can do this by pressing the key **Ctrl** and **+** in VS Code.

Plagiarism Warning

You may discuss with others and refer to any resources. However, any kind of plagiarism will lead to your submission being dismissed. No appeal will be entertained at all.

Late Submission and Penalties

- The submission must be done via eLearning. Other than that (such as telegram, email, google drive, etc.), it will not be entertained at all.
- Programs that CANNOT COMPILE will get a 50% penalty.
- Programs that are submitted late will get a 10% penalty for every day late.

Problem

In this exercise, you will be writing a C++ program that calculates the GPA (Grade Point Average) of a student based on the list of subjects he or she enrolled in. You will be using an Object-Oriented Programming (OOP) approach to write the program.

Notes:

- Write the program in a **separation style**, where the definition of each method is written outside of 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`

The declaration of a class named `Subject` is mostly given in the starter program (main.cpp). Also, the definition of some methods of the class have also been given such as the default constructor, the getter method for credit and grade.

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

1. Define an accessor method to the class named `point()` that determines the point value of the grade earned. For example, if the grade earned is "B+", this method should return 3.33.

2. Define an accessor method to the class named `print()` that prints the subject's information such as the code, name, credit hour, score, etc. onto the screen in a line.

3. Define an overloaded operator in the class for the **'less than'** operator (<) that determines whether a subject is smaller than the other subject. The comparison is done based on the subject's score. This operator should return a Boolean value.

4. Define a regular function that make uses of the operator defined in (3). This function should accept two subjects as parameters and return the smallest one.

5. Define a **friend function** to the class that reads a list of subjects from the user inputs

6. In the main function, write the code to accomplish the following requirements:
   a. Declare an array to hold a list of subjects
   b. Using the function defined in (5), read the inputs and store them into the array.
   c. Using the method defined in (2), print the subject information such as the code, name, score, grade, etc. See example runs in the following figures for the expected results.
   d. Using the function defined in (4), determine the subject that earns the lowest score and print the result. Use the same method from (2) to print the subject. See example runs for the output.

Example Run 1 *(Notes: Bold Text indicates User input)*

```
How many subjects do you want to enter? => 4

Enter info for subject #1:
Subject Code => SECI1013
Subject name => Discrete Structure
Score earned => 75

Enter info for subject #2:
Subject Code => SECJ1013
Subject name => PT 1
Score earned => 56

Enter info for subject #3:
Subject Code => SECP1513
Subject name => TIS
Score earned => 80

Enter info for subject #4:
Subject Code => SECR1013
Subject name => Digital Logic
Score earned => 88


THE RESULT

Subject Code   Subject Name           Credit    Score     Grade     Point     Sub Total

SECI1013       Discrete Structure     3         75        A-        3.67      11.01
SECJ1013       PT 1                   3         56        C+        2.33      6.99
SECP1513       TIS                    3         80        A         4         12
SECR1013       Digital Logic          3         88        A         4         12

TOTAL POINT  : 42
TOTAL CREDIT : 12
GPA          : 3.5

LOWEST SUBJECT :
SECJ1013       PT 1                   3         56        C+        2.33      6.99
```

Example Run 2 *(Notes: Bold Text indicates User input)*

```
How many subjects do you want to enter? => 5

Enter info for subject #1:
Subject Code => SECI2143
Subject name => Statistic
Score earned => 85

Enter info for subject #2:
Subject Code => SECJ1023
Subject name => Prog Tech II
Score earned => 80

Enter info for subject #3:
Subject Code => SECR2033
Subject name => COA
Score earned => 82

Enter info for subject #4:
Subject Code => SECV1113
Subject name => Math for CG
Score earned => 89

Enter info for subject #5:
Subject Code => UHIS1022
Subject name => Current Issues
Score earned => 95


THE RESULT

Subject Code   Subject Name         Credit    Score     Grade     Point     Sub Total

SECI2143       Statistic            3         85        A         4         12
SECJ1023       Prog Tech II         3         80        A         4         12
SECR2033       COA                  3         82        A         4         12
SECV1113       Math for CG          3         89        A         4         12
UHIS1022       Current Issues       2         95        A+        4         8

TOTAL POINT  : 56
TOTAL CREDIT : 14
GPA          : 4

LOWEST SUBJECT :
SECJ1023       Prog Tech II        3         80        A         4         12
```

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 – method `point()`	10
b. Task 2 – method `print()`	10
c. Task 3 – operator <	10
d. Task 4 – function `lower()`	10
e. Task 5 – function `readUserInput()`	10
f. Task 6 – main function	20
2. Pair Programming Session	
a. Video and overall	10
b. Active collaboration	10
c. Both members play both roles Driver and Navigator.	10

Submission

- Deadline: **Saturday, 16 November 2024, 5:00 PM**
- 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. The source code: submit only the **source code (i.e., .cpp)**
  b. The **video link** of your pair programming session with the **student pair names**. Write the link and names in the .cpp 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. **What do we need to show in the video?**

   You should show that you are **doing pair programming** rather than explaining about your code. The video is not meant for presentation.

4. **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.

5. **What if I do this exercise alone? Do I still need to submit the video?**

   In case you got permission to do the exercise alone, you still need to submit the video. You show in the video your progress in doing the exercise. You need to talk about what you are currently coding.

6. **What if we do pair programming 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 recommended to turn on the web camera.