

## Exercise 5 (3%)

### Polymorphism

---

#### 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) (<https://youtu.be/oBraLLybGDA>)
- You and your partner (or group) will be coding collaboratively online.
- You will communicate to each other using Webex, an online meeting software.
- 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 programming 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.
- You may also conduct the pair programming session face to face. However, you still need to record the session.

#### *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.

## 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 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.

## 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.

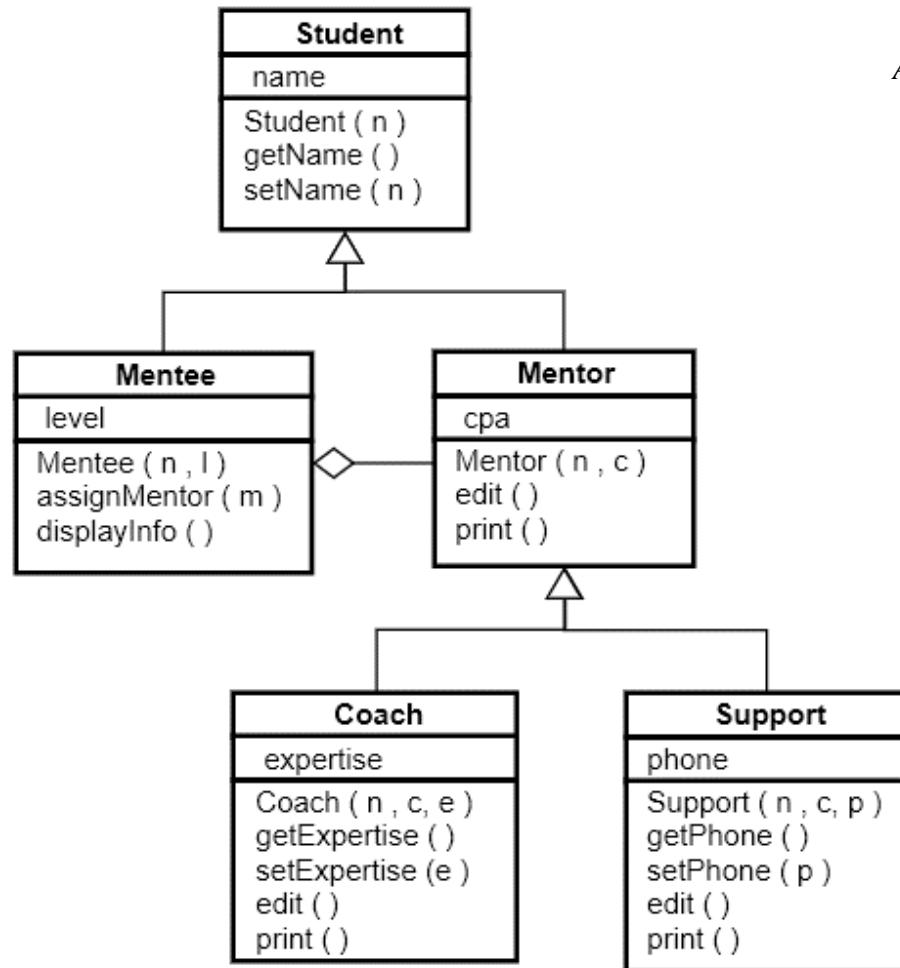
## Late Submission and Penalties

- 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.
- Late submissions will get 10% penalty for every hour late. It will be rounded by ceiling basis. That means, should you submit 1 minute late, it will be considered 1 hour late.

## Question

School of Computing is planning to implement a mentoring programme involving all the students in the school. The programme will be run as follows:

- The students are divided into two categories, mentor and mentee. A mentor is a more knowledgeable student and offers helps to a mentee, a less knowledgeable student.
- Mentors are chosen from those students with good CPAs.
- Mentee's progress is kept track based on their levels, from level 1 to 5.
- There are two approaches of mentoring:
  - Coaching-based: mentors will coach their mentees in particular areas, e.g., "C++ programming", "Game Programming", etc. Therefore, a coach must be expert in that area.
  - Support-based: mentors offer their supports to the mentees without having to meet face to face. Thus, the mentors need to provide their phone numbers.



**Figure 1:** Class diagram for Mentoring System

The school has hired you to develop a computer program to manage the records of the mentors and mentees. The class diagram for the program has been done by a senior software engineer and it is given in Figure 1 above. The classes are described further in Table 1 below.

**Table 1:** Description of the members of each class

<b>Class Members</b>	<b>Description</b>
<b>class Student</b>	
name	Student's name.
Student (...)	A constructor.
getName () and setName (...)	Accessor and mutator to the attribute.
<b>class Mentee</b>	
level	The class attribute indicating the level of progress for each mentee. There are five levels, from 1 to 5, where level 5 indicates mastery level.
Mentee (...)	A constructor.
assignMentor (...)	To assign a mentor to mentee.
displayInfo ()	To display the mentee's information including: <ul style="list-style-type: none"> <li>• name and level,</li> <li>• his/her mentor's details depending the type of the mentor, i.e. either a coach mentor or a support mentor.</li> <li>• If no mentor is assigned yet, then this method will print a message indicating that.</li> </ul>
<b>class Mentor</b>	
cpa	Mentor's cpa.
Mentor (...)	A constructor.
edit ()	To edit mentor information by the user.
print ()	To print the mentor's name and cpa.
<b>class Coach</b>	
expertise	The mentor's area of expertise that he or she can be coaching, e.g., "C++ Programming", "Game Programming", etc.
Coach (...)	A constructor.
getExpertise () and setExpertise (...)	Accessor and mutator to the attribute.
edit ()	To edit mentor's expertise by the user.
print ()	To print the mentor's name, cpa and expertise.

class Support	
phone	The mentor's phone number.
Support (...)	A constructor.
getPhone () and setPhone (...)	Accessor and mutator to the attribute.
edit ()	To edit mentor's phone number by the user.
print ()	To print the mentor's name, cpa and phone.

Based on the class diagram in Figure 1 and specifications in Table 1, you need to modifying code in **exercise5.cpp**. To assist in modifying the code, please consider the followings:

1. Determine the type of relationship formed between the following classes:
  - a. Mentee and Student
  - b. Mentee and Mentor
  - c. Coach and Mentor
  - d. Coach and Student
  - e. Coach and Mentee
  
2. Analyse the class diagram in Figure 1 and its description in Table 1 as well as the program's output in Figure 2. Name TWO (2) methods that later in the the code, will be declared as virtual. Justify your answers.

3. Finally, implement the class diagram by modifying code in **exercise5.cpp**. Complete the following tasks in the program. Your code must follow the class diagram.
  - a. Task 1: Declare all the classes based on the class diagram.
  - b. Task 2: Define all methods for the class `Mentor`
  - c. Task 3: Define all methods for the class `Coach`
  - d. Task 4: Define all methods for the class `Support`
  - e. Task 5: Define all methods for the class `Mentee`
  - f. Task 6: In the main function, an array of mentees and two mentors have been created. Complete the remaining tasks. See Task 6(a) to (h) in the program.

### Output:

The program should produce the output as shown in Figure 2. Note that, **bold texts** indicate user inputs.

#### Screen 1: Displaying the current mentor.

```
===== Menu =====
1. Select first mentor
2. Select second mentor
3. Display selected mentor
4. Edit selected mentor
5. Assign selected mentor to mentee
6. Unassign mentor from mentee
7. Display all mentees
9. Exit

Choose an operation [1-7, or 9] => 3
Current selected mentor: Ahmad Kamal
```

} *mentor1 is the current*

**Screen 2:** Assigning the current mentor to the first and second mentees (i.e index 0 and 1).

```
===== Menu =====
```

1. Select first mentor
2. Select second mentor
3. Display selected mentor
4. Edit selected mentor
5. Assign selected mentor to mentee
6. Unassign mentor from mentee
7. Display all mentees
9. Exit

Choose an operation [1-7, or 9] => **5**

Enter the index of mentee to assign with the current mentor => **0**

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **5**

Enter the index of mentee to assign with the current mentor => **1**

**Screen 3:** Displaying the mentee list.

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **7**

List of mentees:

#1

Mentee's Name : Abdul Samad

Mentee's Level : 2

Mentoring type : Coach-based

Mentor's Name : Ahmad Kamal

Mentor's CPA : 3.87

Mentor's Expertise : Programming

#2

Mentee's Name : Nurdiana

Mentee's Level : 1

Mentoring type : Coach-based

Mentor's Name : Ahmad Kamal

Mentor's CPA : 3.87

Mentor's Expertise : Programming

#3

Mentee's Name : Nurazlan

Mentee's Level : 2

\*\* No mentor yet \*\*

*The last mentee has yet to  
be assigned with a mentor*

**Screen 4:** Changing the current mentor to another and assigning it to the last mentee (index 2).

```
===== Menu =====
```

1. Select first mentor
2. Select second mentor
3. Display selected mentor
4. Edit selected mentor
5. Assign selected mentor to mentee
6. Unassign mentor from mentee
7. Display all mentees
9. Exit

Choose an operation [1-7, or 9] => **2**

Second mentor is selected

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **5**

Enter the index of mentee to assign with the current mentor => **2**

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **7**

List of mentees:

#1

Mentee's Name : Abdul Samad  
Mentee's Level : 2  
Mentoring type : Coach-based  
Mentor's Name : Ahmad Kamal  
Mentor's CPA : 3.87  
Mentor's Expertise : Programming

#2

Mentee's Name : Nurdiana  
Mentee's Level : 1  
Mentoring type : Coach-based  
Mentor's Name : Ahmad Kamal  
Mentor's CPA : 3.87  
Mentor's Expertise : Programming

#3

Mentee's Name : Nurazlan  
Mentee's Level : 2  
Mentoring type : Support-based  
Mentor's Name : Siti Aminah  
Mentor's CPA : 3.98  
Mentor's Phone : 0123456789

*The last mentee now has a  
mentor*



**Screen 5: Changing the expertise info of the first mentor (i.e., mentor1) and the phone number for the second mentor (i.e. mentor2).**

```
===== Menu =====
```

1. Select first mentor
2. Select second mentor
3. Display selected mentor
4. Edit selected mentor
5. Assign selected mentor to mentee
6. Unassign mentor from mentee
7. Display all mentees
9. Exit

Choose an operation [1-7, or 9] => **1**

First mentor is selected

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **4**

Mentor's Information:

Mentoring type : Coach-based  
Mentor's Name : Ahmad Kamal  
Mentor's CPA : 3.87  
Mentor's Expertise : Programming

You can only edit the mentor's expertise

*Current mentor is mentor1  
(a coach mentor). Thus,  
editing will be on coach  
data.*

Enter new expertise => **Mobile and Web Programming**

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **2**

Second mentor is selected

*<The menu is not included here for brevity>*

Choose an operation [1-7, or 9] => **4**

Mentor's Information:

Mentoring type : Support-based  
Mentor's Name : Siti Aminah  
Mentor's CPA : 3.98  
Mentor's Phone : 0123456789

You can only edit the mentor's phone number

*Current mentor is  
changed to mentor2 (a  
support-mentor). Thus,  
editing will be on  
support-mentor data.*

Enter new phone number => **+6013-89001000**

<The menu is not included here for brevity>

Choose an operation [1-7, or 9] => **7**

List of mentees:

#1

Mentee's Name : Abdul Samad

Mentee's Level : 2

Mentoring type : Coach-based

Mentor's Name : Ahmad Kamal

Mentor's CPA : 3.87

Mentor's Expertise : Mobile and Web Programming

} Here, the mentor's  
expertise has  
changed  
accordingly

#2

Mentee's Name : Nurdiana

Mentee's Level : 1

Mentoring type : Coach-based

Mentor's Name : Ahmad Kamal

Mentor's CPA : 3.87

Mentor's Expertise : Mobile and Web Programming

} Here, the mentor's  
expertise has  
changed  
accordingly

#3

Mentee's Name : Nurazlan

Mentee's Level : 2

Mentoring type : Support-based

Mentor's Name : Siti Aminah

Mentor's CPA : 3.98

Mentor's Phone : +6013-89001000

} Here, the mentor's phone number  
has changed accordingly.

## Screen 6: Removing the mentor from the second mentee (i.e., at index 1).

<The menu is not included here for brevity>

Choose an operation [1-7, or 9] => **6**

Enter the index of mentee to remove its mentor => **1**

<The menu is not included here for brevity>

Choose an operation [1-7, or 9] => **7**

List of mentees:

#1

Mentee's Name : Abdul Samad

Mentee's Level : 2

Mentoring type : Coach-based

Mentor's Name : Ahmad Kamal

Mentor's CPA : 3.87

Mentor's Expertise : Mobile and Web Programming

#2

Mentee's Name : Nurdiana

Mentee's Level : 1

\*\* No mentor yet \*\*

} This mentee now does not have a  
mentor.

#3

Mentee's Name : Nurazlan

Mentee's Level : 2

```
Mentoring type : Support-based  
Mentor's Name  : Siti Aminah  
Mentor's CPA   : 3.98  
Mentor's Phone : +6013-89001000
```

**Figure 2:** Program output

## Assessment

This exercise carries **3%** weightage for the final grade of this course. The breakdown weightage is as follows (total 100 marks):

Class declaration, implementation and association. - 25 marks

Inheritance and Polymorphism - Virtual member functions – 30 marks

Overall code implementation, expected output and no error – 25 marks

Video and Pair Programming Session – 20 marks

## Submission

- **Deadline: refer to e-learning**
- Only one member from each pair (or group) needs to do the submission.
- Submission must be done on eLearning.
- You will need to submit TWO (2) items:
  - a. Source code
  - b. The video link of your pair programming session. **Write the link in the header of the source code**