Exercise 4
Association

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

- 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.
- 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.

How To Record the Pair Programming Session

- Use Webex 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.

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

In this exercise you will be adopting the concept of composition to model polynomials. A polynomial is a mathematical function composed of unit expressions called terms. The following is an example of a single-variable polynomial.

$$\mathbf{5x^2 - 2x + 7}$$

Notes: a polynomial may have one, two or more variables. However, in this exercise, the scope is limited only to single-variable polynomials.

Each term of a polynomial contains a **coefficient** and an **exponent**. The polynomial from the above example has three terms and their coefficient and exponent are shown in Table 1.

Table 1

Term	Coefficient	Exponent
$5x^2$	5	2
$-2x$	-2	1
7	7	0

Then, a single-variable polynomial can be modeled with two classes as shown in Figure 1 and their descriptions are given in Table 2.

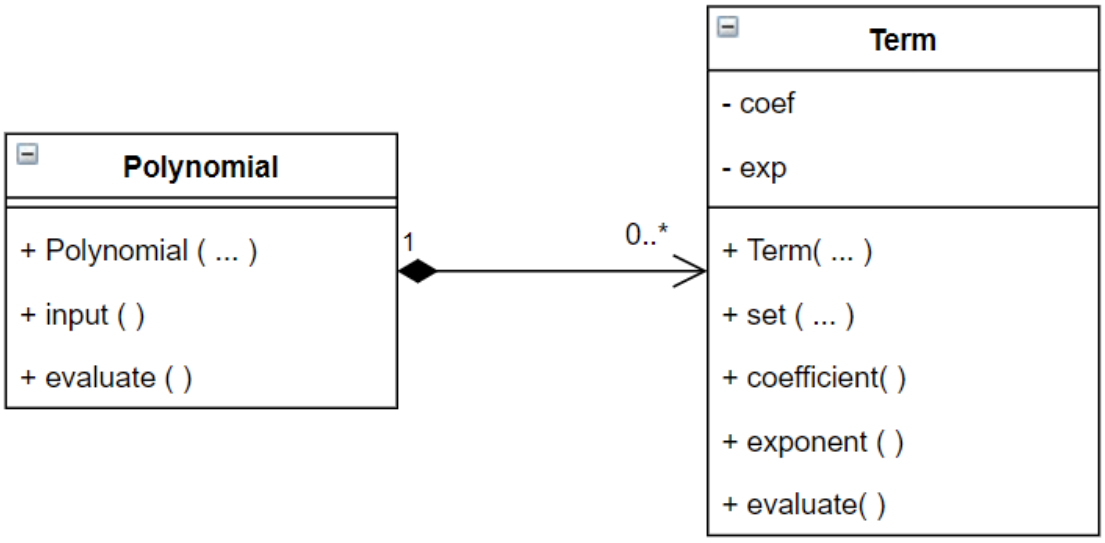

Figure 1

Table 2

Class Members (attributes / methods)	**Description**
class `Term`	
`coef` and `exp`	The attributes for the terms's coefficient and exponent, respectively.
`Term()`	The constructor (s) such as overloaded, default constructor, etc.
`set(c,e)`	sets the term's attributes, `coef` and `exp` respectively.
`coefficient()` and `exponent()`	return the term's attributes, `coef` and `exp` respectively.
`evaluate(x)`	evaluates the term with the value of *x*. For example, if $x = 2$, then the term $5x^2$ will evaluate to $5(2^2) = 20$, and the term $-2x$ will evaluate to $-2(2) = -4$. ***Notes:*** Use the math function, `pow()` to implement this method.
class `Polynomial`	
`input()`	adds the list of terms of a polynomial using user inputs. The user needs to enter the coefficient and exponent for each term.
`evaluate(x)`	evaluates the polynomial by summing up all the terms based on the value of *x*. For example, if $x=2$, then the polynomial $5x^2 - 2x + 7$ will evaluate to $20 – 4 + 7 = 23$.

Based on the classes above, modify the codebase (exercise.cpp) to achieve the goal of the program. Do the following tasks:

Notes: Separate the class declaration and definition in the same file.

1. Implement the class `Term`. Note that, the class's attributes and a constructor have been given in the program. Complete the other methods. Also, you do not need to add additional members for this class.
2. Implement the class `Polynomial`. Add all required attributes and methods to the class.

3. In the main function, write the code to:
    a. create a `Polynomial` object and add terms to the polynomial using user inputs.
    b. print the degree of the polynomial onto the screen.
    c. evaluate the polynomial with different values of *x* from 0 to 5 and print the results onto the screen.

Output:

Expected result from the program is as shown in Figure 2. **Bold** text indicates keyboard inputs.

Run 1 – user enters the polynomial $\mathbf{3}\boldsymbol{x}^{\mathbf{2}}$

```
Enter a polynomial:
      How many terms? => 1
      Enter term #1 (coef and exp) => 3 2

 x    Polynomial value
----  ----------------
0     0
1     3
2     12
3     27
4     48
5     75
```

Run 2 – user enters the polynomial $x - 10$

```
Enter a polynomial:
      How many terms? => 2
      Enter term #1 (coef and exp) => 1 1
      Enter term #2 (coef and exp) => -10 0

 x    Polynomial value
----  ----------------
0     -10
1     -9
2     -8
3     -7
4     -6
5     -5
```

Run 3 – user enters the polynomial $-5x^2 + x^3 + 1 - 7x$

```
Enter a polynomial:
      How many terms? => 4
      Enter term #1 (coef and exp) => -5 2
      Enter term #2 (coef and exp) => 1 3
      Enter term #3 (coef and exp) => 1 0
      Enter term #4 (coef and exp) => -7 1

x     Polynomial value
---- ----------------
0     1
1     -10
2     -25
3     -38
4     -43
5     -34
```

Figure 2

Assessment

This exercise carries **5%** weightage for the final grade of this course. The breakdown weightage is as follows (out of 100 points):

Criteria	Points
The code	
1. Task 1 – class `Term` implementation	
a. `set()`	5
b. `coefficient()`	5
c. `exponent()`	5
d. `evaluate()`	10
2. Task 2 – class `Polynomial` implementation	
a. Attributes	5
b. Constructor	5
c. `input()`	10
d. `evaluate()`	10
3. Task 3 – Main function	
a. create a `Polynomial` object and add terms to the polynomial using user inputs	10
b. evaluate the polynomial with different values of *x* from 0 to 5 and print the results onto the screen.	15
4. Pair Programming Session	
a. Video and overall	10
b. Active collaboration	5
c. Both members play both roles Driver and Navigator.	5

Submission

- Deadline: As specified on eLearning
- Only one member needs to do the submission.
- Submission must be done on eLearning.
- You will need to submit TWO (2) items:
    a. Source code: submit only the source code file, **exercise.cpp**.
    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. You may also choose a partner from different section.

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.