

---

## Exercise 4

### Association, Aggregation & Composition

---

#### 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 can maintain your partner from the previous exercise or change to a new one.
- You and your partner will be coding collaboratively online using VS Code and Live Share.  
[Using Live Share for online collaborative coding](https://youtu.be/s9hfONtUcR8) (<https://youtu.be/s9hfONtUcR8>)
- You will communicate to each other using Google Meet, Webex or any online meeting tool.
- You will record your 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.
- 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 for the pair programming session.

## How To Record the Session

- You can choose Google meet as your online meeting tool and use the feature “**record meeting**” to record your pair programming session.
- Note that a free personal google account does not support the recording feature.
- Try using your student or graduate account from UTM to be able to access the “record meeting” feature.
- Alternatively, you can record your pair programming session locally using software like OBS, PowerPoint, etc. <https://elearning.utm.my/21221/mod/page/view.php?id=26194>

## About the Video

- The video must show that you are coding, communicating, and collaborating with your partner. In this regard, **speak only in English**.
- In the video you should show your VS Code and the output (console).
- You can record the session in a single or multiple videos. If you use multiple videos, put them in a folder, and submit only the folder’s link.
- Set the video file (or folder) permissions with “Anyone can view”.
- 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 (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 10 minutes.

## 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.

$$5x^2 - 2x + 7$$

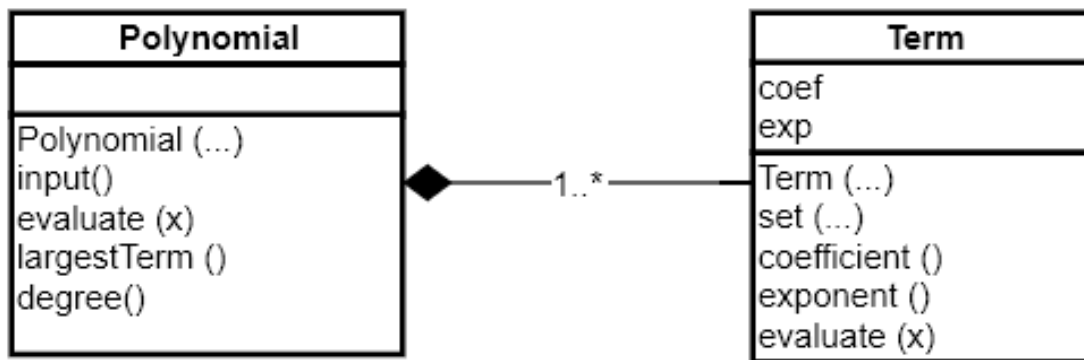
**Notes:** a polynomial may have one, two or more variables. However, in this exercise, the scope is limited only to single-variable polynomials.

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<br>(attributes / methods) | Description  |
|---|--|
| <b>class Term</b>                       |  |
| 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<br>exponent()         | return the term's attributes, coef and exp respectively.   |
| evaluate(x)                             | evaluates the term with the value of x. For example,<br>if $x=2$ , then the term $5x^2$ will evaluate to $5(2^2) = 20$ , and the term $-2x$ will evaluate to $-2(2) = -4$ . <b>Notes:</b> Use the math function, pow() to implement this method. |
| <b>class Polynomial</b>                 |  |
| 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 $20 - 4 + 7 = 23$ .   |
| largestTerm()                           | returns the term whose the largest exponent. For example, the largest term in the polynomial $5x^2 - 2x + 7$ is $5x^2$ , $5 + x^3$ is $x^3$ , $x^2 - 4x^3 + x$ is $-4x^3$ and so on.   |
| degree()                                | returns the degree of a polynomial by taking the largest exponent. For example, the degree of the polynomial $5x^2 - 2x + 7$ is 2, $x^3 - 5$ is 3, $9x$ is 1 and so on.<br><b>Notes:</b> You may want to use the method largestTerm() here.      |

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  $3x^2$

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

Polynomial's degree = 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

Polynomial's degree = 1

  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

Polynomial's degree = 3

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

**Figure 2**

## Assessment

This exercise carries **4%** weightage for the final grade of this course. The breakdown weightage is as follows (out of 100 points):

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

## Submission

- Deadline: Saturday, 1 Jan 2022, 11:59 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., **exercise.cpp** file.
  - b. The video link. Submit the link via eLearning as well.

## FAQs

### 1. Who will be my partner?

Choose your own partner. You may maintain your partner from the previous exercise.

### 2. Can I pair up with someone from a different section?

No.

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

This is only allowed if the number of students in the class is imbalanced. You also need to ask for permission from the lecturer.

### 4. 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.

### 5. 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.

### 6. 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.

### 7. 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 recommended to turn on the web camera. Keep in mind that you keep following the SOP about COVID-19 when working in a face-to-face environment.