

Exercise 5

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 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 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.

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 writing a C++ program that implements the concept of polymorphism. The program will do some manipulations on two types of shapes, circles, and rectangles, including resizing and moving the shape. The program will show a list of shapes on the screen randomly. However, the user can only manipulate one shape at a time. In this case, he or she first needs to choose a shape from the list using a mouse click. Table 1 shows the list the operations and their commands. Expected result of the program is given in several videos that come with this exercise.

Table 1: The commands for the user

User Interaction	Operation
Mouse click	To choose a shape that the user will be manipulating or working on.
Keyboard press + or -	To enlarge or shrink the selected shape.
Arrow keys	To move the object accordingly, i.e, to the left, right, up or down.

You are expected to write the program by separating class specification and implementation into their dedicated files, i.e. .hpp and .cpp respectively. In fact, the separation has been done for you in the codebase that comes with this exercise. Besides, the code for some parts of the program has been provided including the basic structure of each class, and the main loop in the main function. Modify the codebase to achieve the goal of the program. Note that you **must implement the concept of polymorphism** to write the program.

Formulas

Below are the formulas that you may need to use in your code:

- **Determining whether a mouse click is inside a circle.**

Assuming a mouse click at location (mx, my) , the mouse is considered inside the circle if its distance from the circle's center (cx, cy) , d is smaller than or equal to the circle's radius, r .

i.e, $d \leq r$

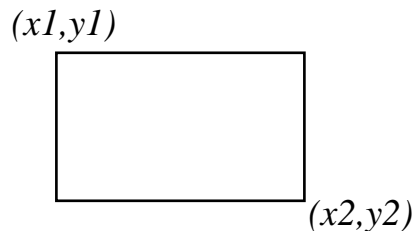
The distance can be determined by Pythagorean theorem:

$$d = \sqrt{(mx - cx)^2 + (my - cy)^2}$$

- **Determining whether a mouse click is inside a rectangle.**

Assuming a mouse click at location (mx, my) , the mouse is considered inside the rectangle if the following conditions are met. $x1 \leq mx \leq x2$ and $y1 \leq my \leq y2$

Where $(x1, y1)$ and $(x2, y2)$ are the top-left and bottom-right corners of the rectangle as shown in the following figure.



Resources

1. WinBGIm documentation

<https://home.cs.colorado.edu/~main/cs1300/doc/bgi/>

See the following functions. You may need using them in this exercise.

- `circle()` – to draw an outlined circle
- `rectangle()` – to draw an outlined rectangle
- `setcolor()` – to specify the outline color
- `getch()` – to read a character from the keyboard
- `getmouseclick()` – to read input from the mouse click

2. How to use random numbers in C++

<https://www.cplusplus.com/reference/cstdlib/rand/>

Assessment

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

Criteria	Points
1. Class Shape	
a. Class declaration (shape.hpp), attributes and methods	10
b. Class implementation (shape.cpp)	10
<i>Notes, make this class abstract</i>	
2. Class Circle	
a. Class declaration (circle.hpp), attributes and methods	5
b. Class implementation (circle.cpp)	10
3. Class Rect	
a. Class declaration (rect.hpp), attributes and methods	5
b. Class implementation (rect.cpp)	10
4. The main function (main.cpp)	
a. Show a list of shapes randomly.	
<i>Random here means, each element of the list is random, i.e., whether a circle or rectangle. Their initial locations are also random. At the beginning no shape is selected.</i>	10
b. Select a shape with a mouse click.	
<i>The selected shape will turn highlighted (for example draw the outline with yellow color). Clicking on the selected shape will turn it back to unselected.</i>	10
c. Resize (enlarge or shrink) the selected shape's size by pressing the plus (+) or minus (−) key, respectively.	
<i>Decide how large or small the selected object will be resized on your own. This can be hard-coded.</i>	5
d. Move the selected shapes to the left, right, up or bottom with the arrow keys.	
	5
5. 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 stated in e-Learning
- Only one member from each pair needs to do the submission.
- Submission must be done on eLearning.
- You will need to submit TWO (2) items:
 - a. Source code: compress all source code files into a **zip file**. Submit only the zip file.
 - b. The video link of your pair programming session. **Write the link in the source code (in the 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. 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.