Exercise 4
Polymorphism

Duration : 1 hour 45 minutes

Objectives
This exercise aims at implementing the following concept:
- Polymorphism

Exercise Materials
- Program templates are provided for this exercise. Please download from the elearning and extract the ZIP file to your local drive.
- You have the choice to use Microsoft **VS Code**, **Dev C++** or any other IDEs to write the code for this exercise.
- If you choose VS Code, use debug **"WinBGIm program"** to run the program.
- If you use Dev C++, you need to setup the IDE so that it can use WinBGIm library.

Deliverable Item
- Only the **source code** file is needed for the submission (i.e., **exercise4.cpp**).
- You must submit your source code at elearning.

Plagiarism Policy
- Discussions among the students are still possible during the exercise session.
- However, all works must be done individually.
- Any kind of plagiarism (e.g., copy and paste code by any mean) would lead to disqualification of submissions for both parties (i.e., students that copy others' code and students that give their code to others).

Late Submission Policy
- 5% deduction for every 5 minutes late.

Case Study

In this exercise you will be creating a program which displays two geometrical objects, a circle and rectangle on the screen. The user is then allowed to control the objects such as changing the object's color and size, and moving the objects to other locations. Here, the user interacts with the program using keyboard commands. Table 1 belows shows the list of commands that the program should provide.

Table 1: The commands

Keyboard key	**Operation**
C	To choose or select the circle object.
R	To choose the rectangle object.
B	To change the color of the object to blue.
Y	To change the color of the object to yellow.
L	To enlarge the object.
S	To shrink the object.
Arrow keys	To move the object to the left, right, up and down, accordingly.

Notes: Run the executable file provided, **expected_result.exe**, to see how the program should look like.

Figure 1 shows the class diagram depicting the all the classes involved in this program and their relationships.

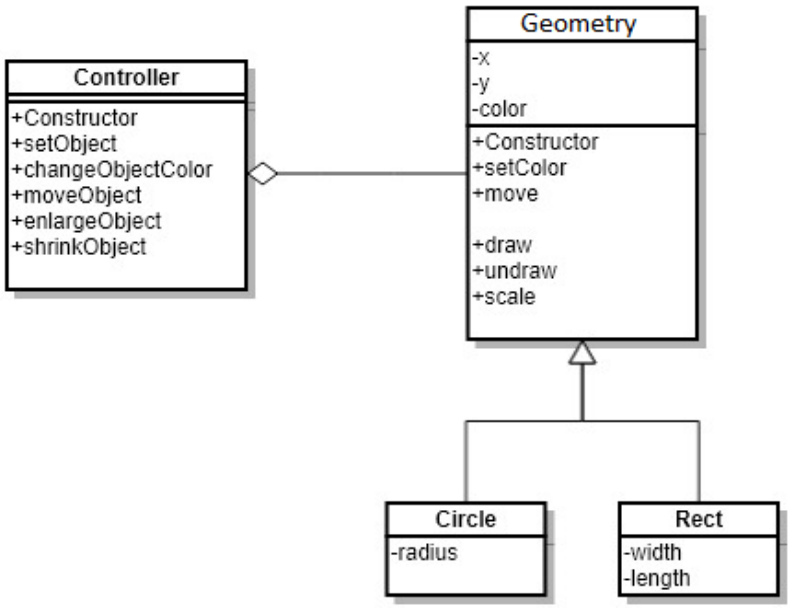

Figure 1: Class diagram for geometrical object manipulations

For the class `Controller`, as the name implies, it is the main class to control the manipulation of the geometrical objects. In this case, the class provides several methods for the manipulations as shown in Figure 1.

Tasks

Based on the requirements given above and the class diagram in Figure 1, modify the the provided template source code file, **exercise4.cpp** to achieve the goal of the program. The implementation for the class `Geometry` and some other parts of the program have been given in the program file. Complete the following tasks in the program:

1. Complete the implementation for the class `Circle` including its relationship to the class `Geometry`.
2. Complete the implementation for the class `Rect` including its relationship to the class `Geometry`.
3. Complete the implementation for the class `Controller` including its relationship to the class `Geometry`.
4. Complete the implementation for the main function.

Detailed instructions are stated inside the program file.

Note that, you need to implement the concept of polymorphism for this program