

---

## Tutorial 9

### Polymorphism

---

#### Program 1

Develop an application that allows the user to manipulate two types of shapes (rectangle and circle). The user needs to choose which shape he or she wants to work on For example: by pressing 'C' for circle and 'R' for rectangle.

Once selected a shape, the user can manipulate the shape on two operations:

- Move the object (up, down, left or right) - using arrow keys
- Resize the object (enlarge or shrin the size) - using + or - keys.

For convenience two plain classes, `Circle` and `Rect` have been provided. The classes are organized such that they do not form any relationship. You are not expected to modify the class. Complete program1 by modifying only in the main function.

#### Program 2

Rewrite the same program you did above, but this time using polymorphism.

### Program 3

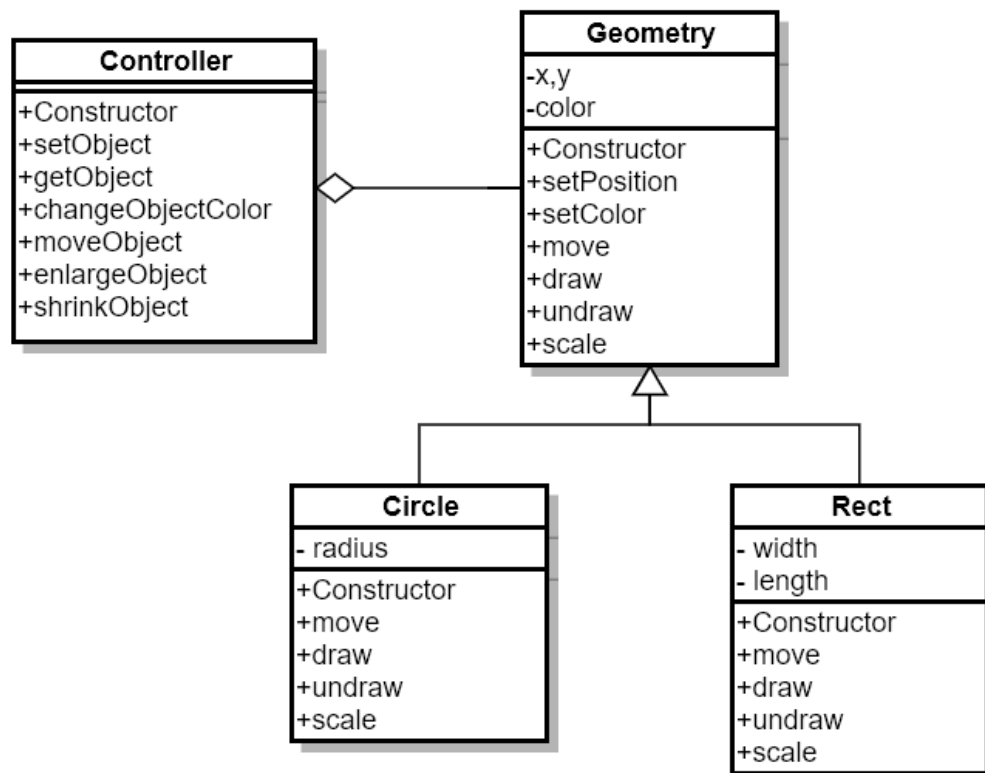
In this tutorial you will be implementing the concept of polymorphism in a program that displays two geometrical objects, a circle and rectangle on the screen. The user is then allowed to control the objects such as changing the object's color and size, and moving the objects to other locations. Here, 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.

**Note:** 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, **tutorial91.cpp** to achieve the goal of the program. Complete the following tasks in the program:

1. Complete the declaration of the class `Geometry` and answer some questions. See tasks 1(a) to 1(d) in the program.
2. Complete the declaration of the class `Circle` and answer some questions. See tasks 2(a) to 2(c) in the program.
3. Complete the declaration of the class `Rect`. See tasks 3(a) and 3(b) in the program.
4. Complete the declaration of the class `Controller` and answer some questions. See tasks 4(a) to 4(c) in the program.
5. Complete the definition of class `Circle`. See tasks 5(a) and 5(b) in the program.
6. Complete the definition of class `Rect`. See tasks 6(a) to 6(d) in the program.
7. Complete the definition of class `Controller`. See tasks 7(a) to 7(g) in the program.
8. Complete the implementation for the main function. See task 8 in the program.
9. Add ONE(1) additional feature to the program. Detailed instruction is given in the program, see task 9.

**Note:** Run the executable file provided, **expected\_result.exe**, to see some examples of features that you may add to the program.

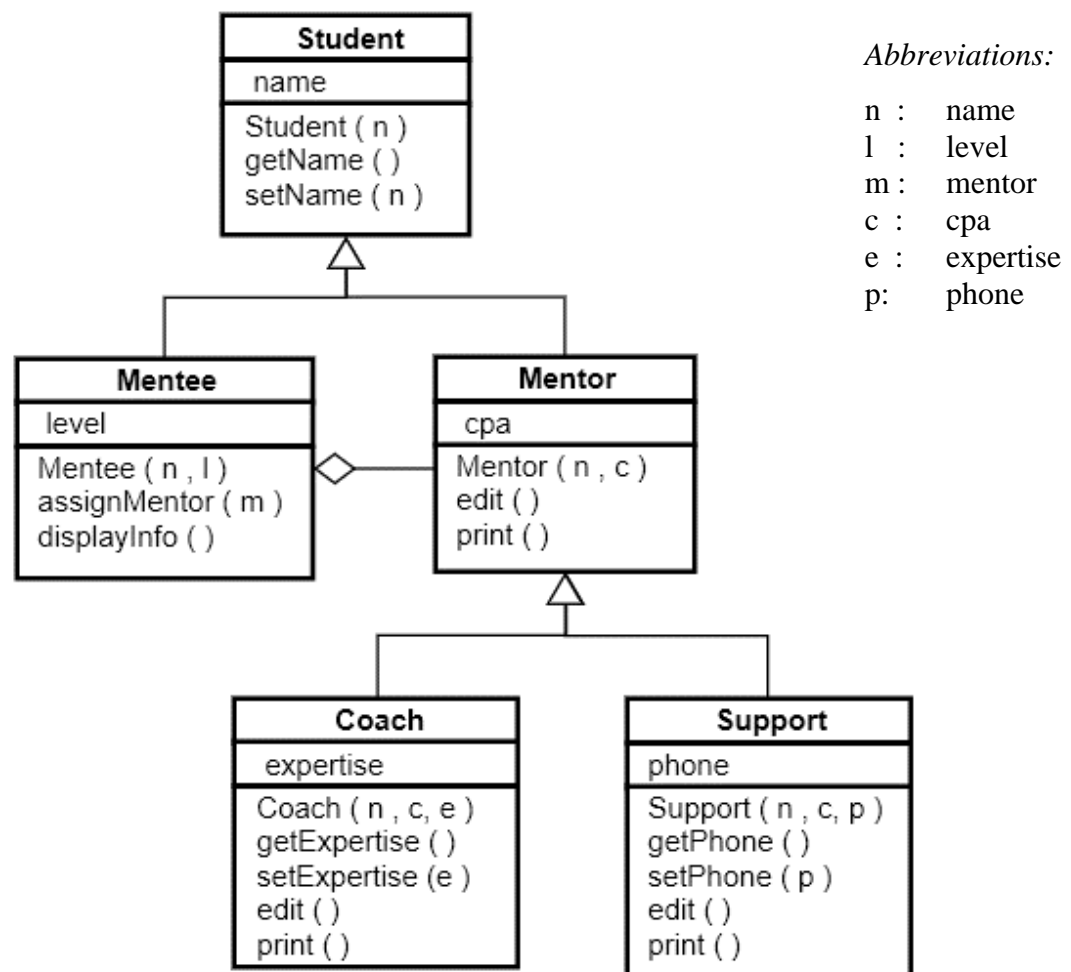
Keyboard key Or Mouse click	Operation
1	To move the selected object to the middle of the screen
2	To animate the selected object: blinking effect
3	To animate the selected object: moving
4	To animate the selected object: resizing
Mouse click	To position the object to the location where the mouse is clicked on.

## Program 4

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 are 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 4.1:** 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 4.1 above. The classes are described further in Table 1 below.

**Table 4.1:** 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, answer the questions (a) to (c) below.

**Note:** As for questions 1 and 2, write your answers in a **separate text file**. Indicate the question numbers in your answers.

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 4.1 and its description in Table 4.1 as well as the program's output in Figure 4.2. Name TWO (2) methods that later in the code, will be declared as virtual. Justify your answers.

3. Finally, implement the class diagram by modifying code in **exercise3.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 mentor*

**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

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

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:
```

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

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

```

#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 4.2: Program output**