|  |  |
| --- | --- |
| **Programming 2**  Diploma in IT / DS / CSF  Year 1 (2021/22) Semester 2 | Week **7** |
| **2 hours** |
| **Practical 7 : Polymorphism** | |

**Objectives**

At the end of this practical, the students should be able to:

* Write programs incorporating polymorphism
* Use superclass variable to reference objects of its subclasses
* Use casting

|  |
| --- |
| **IMPORTANT**   * Create a folder, **week07.** * Make a copy of the project, **Snnnnnnnn\_ShapeApp**, in the **Week06** folder and place in your **Week07** folder *(note:* ***Snnnnnnnn*** *is your Student Number)*. * At the end of the session, copy the folder **Week07** folder (which contains all your work) to PRG2 network folder: **\\ictspace.ict.np.edu.sg\PRG2**. |

Do the following modifications to the Project **ShapeApp**:

1. Based on the class diagram given in Appendix 1, implement the **Square** class.
2. Modify the method that displays menu so that the menu display is as shown below:

|  |
| --- |
| -----------Menu------------  [1] List all the shapes  [2] Display the areas of the shapes  [3] Display the perimeters of the shapes  [4] Change the sizes of the shapes  [5] Add a new circle  [6] Delete a circle  [7] Display shapes sorted by area  [0] Exit  ---------------------------  Enter your option: |

1. In the **Program.cs**, do the following:
2. Modify the List to **shapeList** for storing Shape objects.
3. Rename the **InitCircleList()** method to **InitShapeList()** and modify it to do the following:
   1. create a red circle (object) with radius 20.0 and assign it to a variable, shape1.
   2. create a red square (object) with length 10.0 and assign it to a variable, shape2.
   3. create a green circle (object) with radius 10.0 and assign it to a variable, shape3.
   4. create a green square (object) with length 20.0 and assign it to a variable, shape4.
   5. create a blue circle (object) with radius 30.0 and assign it to a variable, shape5.
   6. create a blue square (object) with length 30.0 and assign it to a variable, shape6.
   7. add these 6 objects to **shapeList**.
4. Modify the code for each option accordingly as described below:
   1. Option 1: List all the shapes

Display the details of all the Shape objects in shapeList. Display an error message if the list is empty.

|  |
| --- |
| Enter your option: 1  [1] Type: Circle Color: Red Radius: 20  [2] Type: Square Color: Red Length: 10  [3] Type: Circle Color: Green Radius: 10  [4] Type: Square Color: Green Length: 20  [5] Type: Circle Color: Blue Radius: 30  [6] Type: Square Color: Blue Length: 30 |

* 1. Option 2: Display the areas of the shapes

Display the attributes and area of the Shape objects in shapeList.

|  |
| --- |
| Enter your option: 2  Type: Circle Color: Red Radius: 20 Area: 1256.64  Type: Square Color: Red Length: 10 Area: 100.00  Type: Circle Color: Green Radius: 10 Area: 314.16  Type: Square Color: Green Length: 20 Area: 400.00  Type: Circle Color: Blue Radius: 30 Area: 2827.43  Type: Square Color: Blue Length: 30 Area: 900.00 |

* 1. Option 3: Display the perimeters of the shapes

Display the attributes and perimeter of the Shape objects in shapeList.

|  |
| --- |
| Enter your option: 3  Type: Circle Color: Red Radius: 20 Perimeter: 125.66  Type: Square Color: Red Length: 10 Perimeter: 40.00  Type: Circle Color: Green Radius: 10 Perimeter: 62.83  Type: Square Color: Green Length: 20 Perimeter: 80.00  Type: Circle Color: Blue Radius: 30 Perimeter: 188.50  Type: Square Color: Blue Length: 30 Perimeter: 120.00 |

* 1. Option 4: Change the sizes of the shapes

Increase the size (radius or length) of each Shape object by 5 units and display the details of all the objects.

|  |
| --- |
| Enter your option: 4  [1] Type: Circle Color: Red Radius: 25  [2] Type: Square Color: Red Length: 15  [3] Type: Circle Color: Green Radius: 15  [4] Type: Square Color: Green Length: 25  [5] Type: Circle Color: Blue Radius: 35  [6] Type: Square Color: Blue Length: 35 |

* 1. Option 5: Add a new circle

Add a Circle object to shapeList.

|  |
| --- |
| Enter your option: 5  Circle color: Red  Circle radius: 25  New Red circle with radius 25cm added. |

* 1. Option 6: Delete a circle

Delete the last Circle object in shapeList. Display an error message if the list is empty or the list does not contain any Circle object.

|  |
| --- |
| Enter your option: 6  [1] Type: Circle Color: Red Radius: 20  [2] Type: Square Color: Red Length: 10  [3] Type: Circle Color: Green Radius: 10  [4] Type: Square Color: Green Length: 20  [5] Type: Square Color: Blue Length: 30  Circle removed. |

**Advanced**

Option 7: Display shapes sorted by area

|  |
| --- |
| Enter your option: 7  Type: Square Color: Red Length: 10 Area: 100.00  Type: Circle Color: Green Radius: 10 Area: 314.16  Type: Square Color: Green Length: 20 Area: 400.00  Type: Square Color: Blue Length: 30 Area: 900.00  Type: Circle Color: Red Radius: 20 Area: 1256.64  Type: Circle Color: Blue Radius: 30 Area: 2827.43 |

*The output is obtained before option 6 is called.*

**Hint:**

1. Modify the **Shape** class to implement the **interface** below so that the objects of **Shape** class can be sorted based on its area.

|  |
| --- |
| interface IComparable<T>  {  int CompareTo(T obj);  } |

The return value of CompareTo() method is:

1 if the object is bigger than the other object

0 if they are the same

-1 if the other object is bigger

1. Implement the interface



1. Implement the **CompareTo()** method in the Shape class:



1. Modify the code in option 7 to call the sort method on the list object and display the sorted list.

Appendix 1 – *Class Diagram*

|  |
| --- |
| ***Shape*** |
| -type:string  -color:string |
| +Shape()  +Shape(string,string)  *+FindArea():double*  *+FindPerimeter():double*  +ToString():string |

|  |  |  |
| --- | --- | --- |
| **Circle** |  | **Square** |
| -radius:double |  | -length:double |
| +Circle()  +Circle(string,double)  +FindArea():double  +FindPerimeter():double  +ToString():string |  | +Square()  +Square(string,double)  +FindArea():double  +FindPerimeter():double  +ToString():string |

**Plagiarism Warning:**

**If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical. Disciplinary action may also be taken.**

**Similar action will be taken for student who allows other student(s) to copy his/her work.**