**BCS 345 Lab – Polymorphism and Interfaces**

***Overview***

Write a program that uses polymorphism and interfaces.

***Part 1 – Driver Class***

Create a class called Driver that has a main method.

***Part 2 – Shape Class***

Create an abstract class named Shape. It should contain a member variable for area that is a double. The member variable should be accessible to all derived classes.

Add an abstract method named CalculateArea. It should have a void return type and take 0 parameters.

***Part 3 – Circle Class***

Create a class named Circle that is derived from Shape. It should contain a member variable for radius that is a double. Add a constructor that takes the radius as a parameter and sets the member variable radius to that value.

Override the CalculateArea method. It should set the area based on the radius. Use the following formula: 3.14 \* radius \*radius

***Part 4 – Rectangle Class***

Create a class named Rectangle that is derived from Shape. It should contain member variables for length and width that are double. Add a constructor that takes the length and width as parameters and sets the corresponding member variables.

Override the CalculateArea method. It should set the area based on the length and width. Use the following formula: length \* width

***Part 5 – Main Code***

In main, create an array of Shape that contains 4 elements. Put instances of Circle and Rectangle in the array (two instance of each class should be in the array).

Write a loop that calls CalculateArea on each array element.

***Part 6 – IDisplayable***

Create an interface.

***IDisplayable Interface***

*Methods: void Display();*

*Methods that implement the Display method should show the area and all other member variable data on the screen.*

***Part 7 – Update Circle and Rectangle***

Update the Circle and Rectangle classes so that they implement the IDisplayable interface.

***Part 8 – Driver ShowAllData Method***

Add a static method named ShowAllData to the Driver class. It should take an array of IDisplayable as a parameter. This method should contain a loop that calls Display() on all elements of the array.

***Part 9 – Update Main Code***

In main, create an array of IDisplayable that contains 4 elements. Put instances of Circle and Rectangle in the array (two instance of each class should be in the array).

Add a call to ShowAllData after you create the IDisplayable array.

Run the program. Are the areas correct in the output? Can you call CalculateArea on elements of this array?

***Part 10 – Update Implementing IDisplayable***

Remove the code in the class headers of Circle and Rectangle that indicate that those classes implement the IDisplayable interface.

Add code to the Shape class header to make it implement IDisplayable.

***Part 11 – Add Circumference Calculations***

Add a member variable to for circumference to the Shape class. The member variable should be accessible to all derived classes.

Add an abstract method to the Shape class named CalculateCircumference (void return type and no parameters). Update derived classes as necessary to handle the new functionality.

Circle Circumference – 2 \* 3.14 \* radius

Rectangle Circumference – 2 \* length + 2 \* width

Update Display methods where necessary to also show the circumference.

Update main so that it calls CalculateCircumference (in addition to CalculateArea) on all elements of the Shape array.

Add a call to ShowAllData that takes the Shape array. We are allowed to do this now because Shape now implements the IDisplayable interface.