# Task 1

1. **BasicShape interface:**

Design an interface called BasicShape with the following members:

- `calcArea()`: A method that should be implemented by any class that implements this interface. This method calculates the area of the shape.

- `display()`: A method that should be implemented by any class that implements this interface. This method displays information about the shape.

**2. Displayable interface:**

Design an interface called Displayable with the following members:

- `displayWithPrecision(int precision)`: A method that should be implemented by any class that implements this interface. This method displays information about the shape with the specified precision.

**3. ShapeWithArea abstract class:**

Design an abstract class called ShapeWithArea. This class should implement the BasicShape interface and provide the following member:

- area: a double used to hold the shape's area.

Public Member Functions:

- `getArea()`: Returns the value of the member variable "area."

**4. Circle class:**

Design a class called Circle, which extends the ShapeWithArea abstract class and implements the BasicShape and Displayable interfaces, to represent a circle. The Circle class should have the following members:

Private Member Variables:

- centerX: a long integer used to hold the x-coordinate of the circle's center.

- centerY: a long integer used to hold the y-coordinate of the circle's center.

- radius: a double used to hold the circle's radius.

Public Member Functions:

- Constructor: Accepts values for centerX, centerY, and radius. Calls the `calcArea()` method to calculate and store the area.

- `getCenterX()`: Returns the value of the member variable "centerX."

- `getCenterY()`: Returns the value of the member variable "centerY."

- `calcArea()`: Overrides the `calcArea()` method from the BasicShape interface to calculate the area of the circle (area = 3.14159 * radius * radius) and stores the result in the inherited member variable "area."

- `display()`: Implements the `display()` method from the BasicShape interface to show information about the circle with default precision.

- `displayWithPrecision(int precision)`: Implements the `displayWithPrecision(int precision)` method from the Displayable interface to show information about the circle with the specified precision.

**5. Rectangle class:**

Design a class called Rectangle, which extends the ShapeWithArea abstract class and implements the BasicShape and Displayable interfaces, to represent a rectangle. The Rectangle class should have the following members:

Private Member Variables:

- width: a long integer used to hold the width of the rectangle.

- length: a long integer used to hold the length of the rectangle.

Public Member Functions:

- Constructor: Accepts values for width and length. Calls the `calcArea()` method to calculate and store the area.

- `getWidth()`: Returns the value of the member variable "width."

- `getLength()`: Returns the value of the member variable "length."

- `calcArea()`: Overrides the `calcArea()` method from the BasicShape interface to calculate the area of the rectangle (area = length * width) and stores the result in the inherited member variable "area."

- `display()`: Implements the `display()` method from the BasicShape interface to show information about the rectangle with default precision.

- `displayWithPrecision(int precision)`: Implements the `displayWithPrecision(int precision)` method from the Displayable interface to show information about the rectangle with the specified precision.

**6. A Driver Class:**

Create a driver class with main method that instantiates objects of Circle and Rectangle classes. Demonstrate that each object properly calculates and displays its area and additional information with default precision and custom precision using the `display()` and `displayWithPrecision(int precision)` methods from the BasicShape and Displayable interfaces, respectively.