```
namespace backendDay6
{
    0 references
    internal class Program
    {
        5 references
        public class Shape
        {
            5 references
            public virtual void CalculateArea()
            {
                Console.WriteLine("Calculating the area of a shape...");
            }
        }

        6 references
        public static class InputHelper
        {
            6 references
            public static double ReadInput(string prompt)
            {
                double value;
                bool valid;
                do
                {
                    Console.Write(prompt);
                    valid = double.TryParse(Console.ReadLine(), out value) && value > 0;
                    if (!valid)
                    {
                        Console.WriteLine("Value must be a number greater than 0.");
                    }
                } while (!valid);

                return value;
            }
        }
```

```
1 reference
public class Circle : Shape
{
    2 references
    public override void CalculateArea()
    {
        Console.WriteLine("Calculating the area of a circle");
        double radius = InputHelper.ReadInput("Please enter the radius: ");
        double area = Math.Pow(radius, 2) * Math.PI;

        Console.WriteLine($"The area of a circle with a radius of {radius} is {area:F2}");
    }
}

1 reference
public class Rectangle : Shape
{
    2 references
    public override void CalculateArea()
    {
        Console.WriteLine("Calculating the area of a rectangle");
        double length = InputHelper.ReadInput("Enter length: ");
        double width = InputHelper.ReadInput("Enter width: ");

        double area = length * width;
        Console.WriteLine($"The area of a rectangle with a lenght of {length} and width of {width} = {area:F2}");

    }
}

1 reference
public class Square : Shape
{
    2 references
    public override void CalculateArea()
    {
        Console.WriteLine("Calculating the area of a square");
        double sides = InputHelper.ReadInput("Enter side: ");

        double area = sides * sides;
        Console.WriteLine($"The area of a square with a side of {sides} = {area:F2}");
    }
}
```

```
1 reference
public class Triangle : Shape
{
    2 references
    public override void CalculateArea()
    {
        Console.WriteLine("Calculating the area of a triangle");
        double triangleBase = InputHelper.ReadInput("Enter base: ");
        double height = InputHelper.ReadInput("Enter height: ");

        double area = (triangleBase * height)/2;
        Console.WriteLine($"The area of a triangle with a base of {triangleBase} and height of {height} = {area:F2}");
    }
}
```

```
        static void Main(string[] args)
        {
            int operation;
            bool valid;
            string useAgain;

            do
            {
                do
                {
                    Console.WriteLine("===========\t SHAPE AREA CALCULATOR! \t===========");
                    Console.WriteLine("Please Select a Shape!");
                    Console.WriteLine("1. Circle\n2. Rectangle\n3. Square\n4. Triangle\n5. Exit");
                    Console.Write("Type your option: ");
                    valid = int.TryParse(Console.ReadLine(), out operation) && operation > 0 && operation < 6;
                    if (!valid)
                    {
                        Console.WriteLine("Please select from the choices only.");
                    }
                } while (!valid);

                Shape shape = null;

                switch (operation)
                {
                    case 1:
                        shape = new Circle();
                        break;
                    case 2:
                        shape = new Rectangle();
                        break;
                    case 3:
                        shape = new Square();
                        break;
                    case 4:
                        shape = new Triangle();
                        break;
                    case 5:
                        Console.WriteLine("Exiting the program. Goodbye!");
                        return;
                }

                shape?.CalculateArea(); // checks if null pag hindi di priprint

                Console.Write("Do you want to calculate another shape area? (yes or no): ");
                useAgain = Console.ReadLine().ToLower();

            } while (useAgain == "yes");
        }
    }
}
```

OUTPUTS

```
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 1
Calculating the area of a circle
Please enter the radius: 2
The area of a circle with a radius of 2 is 12.57
Do you want to calculate another shape area? (yes or no): yes
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 2
Calculating the area of a rectangle
Enter length: 5
Enter width: 10
The area of a rectangle with a lenght of 5 and width of 10 = 50.00
Do you want to calculate another shape area? (yes or no): yes
```

```
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 3
Calculating the area of a square
Enter side: 5
The area of a square with a side of 5 = 25.00
Do you want to calculate another shape area? (yes or no): yes
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 4
Calculating the area of a triangle
Enter base: 2
Enter height: 4
The area of a triangle with a base of 2 and height of 4 = 4.00
Do you want to calculate another shape area? (yes or no):
```

ERROR HANDLING

```
Please select from the choices only.
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 7
Please select from the choices only.
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: -2
Please select from the choices only.
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
```

```
Type your option: 1
Calculating the area of a circle
Please enter the radius:
Value must be a number greater than 0.
Please enter the radius:
Value must be a number greater than 0.
Please enter the radius: -2
Value must be a number greater than 0.
Please enter the radius: -.32
Value must be a number greater than 0.
Please enter the radius:
```

IF CHOICE = NO

```
Do you want to calculate another shape area? (yes or no): no

D:\NICO\WPH\backendDay6\bin\Debug\net8.0\backendDay6.exe (process 580) exited with code 0 (0x0).
Press any key to close this window . . .
```

```
==========      SHAPE AREA CALCULATOR!         ==========
Please Select a Shape!
1. Circle
2. Rectangle
3. Square
4. Triangle
5. Exit
Type your option: 5
Exiting the program. Goodbye!
```

REFLECTION

In this activity, I used polymorphism by creating a base class called Shape that has a method named CalculateArea().I created separate classes for each shape like Circle, Rectangle, Square, and Triangle, and used the override keyword in each one to create their own version of CalculateArea() using the correct formula. This allows me to use a Shape variable in the Main method and still call the correct method depending on the user's choice. That is how I applied polymorphism by using the same method name but with different behaviors depending on the object.

For user input, I made a static class called InputHelper with a method called ReadInput(). It uses a do while loop that keeps asking the user until they enter a valid number that is greater than zero. This helped make sure the program does not crash or accept wrong input. I also used variables like operation to store the user's selected shape, and inside each shape class I stored the needed values like radius, length, width, base, or height. After the user chooses a shape from the menu, I used a switch statement to create the right shape object and store it in a Shape variable. Then outside the switch, I called shape.CalculateArea() which automatically runs the correct version of the method depending on what shape was created.

One thing I learned from this activity is how useful polymorphism is. At first, I thought I had to call each shape method separately, but now I understand that one method call can do all the work if the setup is correct. I did get a bit confused at first when I called the method inside each case, but after I moved the method call outside the switch, the code became much cleaner. I also found it helpful to have a separate class just for input so I did not have to repeat code every time I needed user input. This activity really helped me understand how object-oriented programming makes code more flexible, organized, and easier to manage.