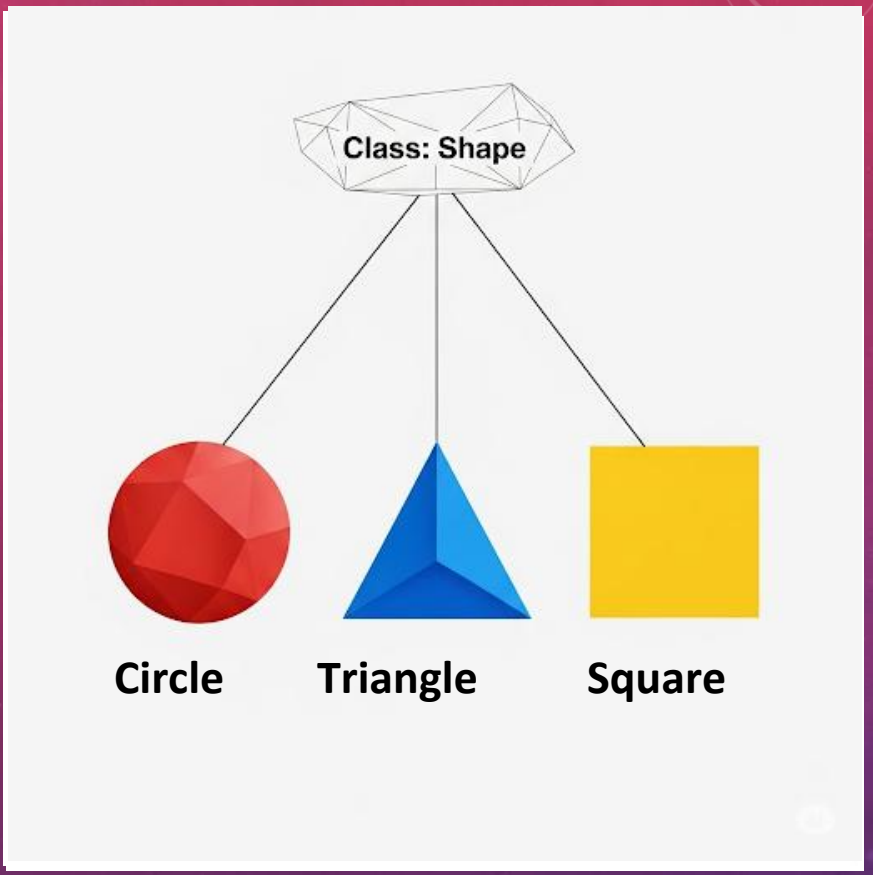
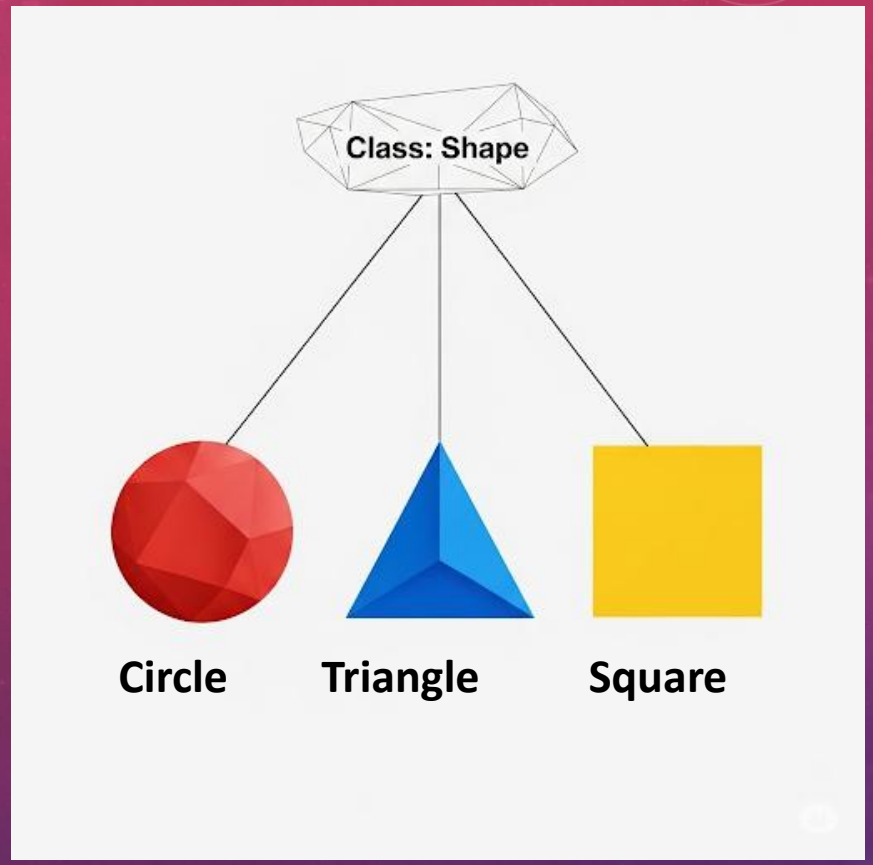


Class: A template that defines what an object will have and what it can do.
Object: A specific instance of a class, a real item created from that template.

Abstract Class: An abstract class is a template that cannot be created directly, used to enforce that only specific shapes like Square can be made.

OBJECT-ORIENTED PROGRAMMING (OOP) IN C#

OOP in C# is a way to write code by thinking about things as "objects" from the real world. You create templates called "classes" to define what an object is and what it can do, and then you build many different objects from those templates.



Class: A template that defines what an object will have and what it can do.
Object: A specific instance of a class, a real item created from that template.

Properties: The attributes or data an object **has** (e.g., color, size, name).

Methods: The actions or behaviors an object **can do** (e.g., draw, calculate, move).

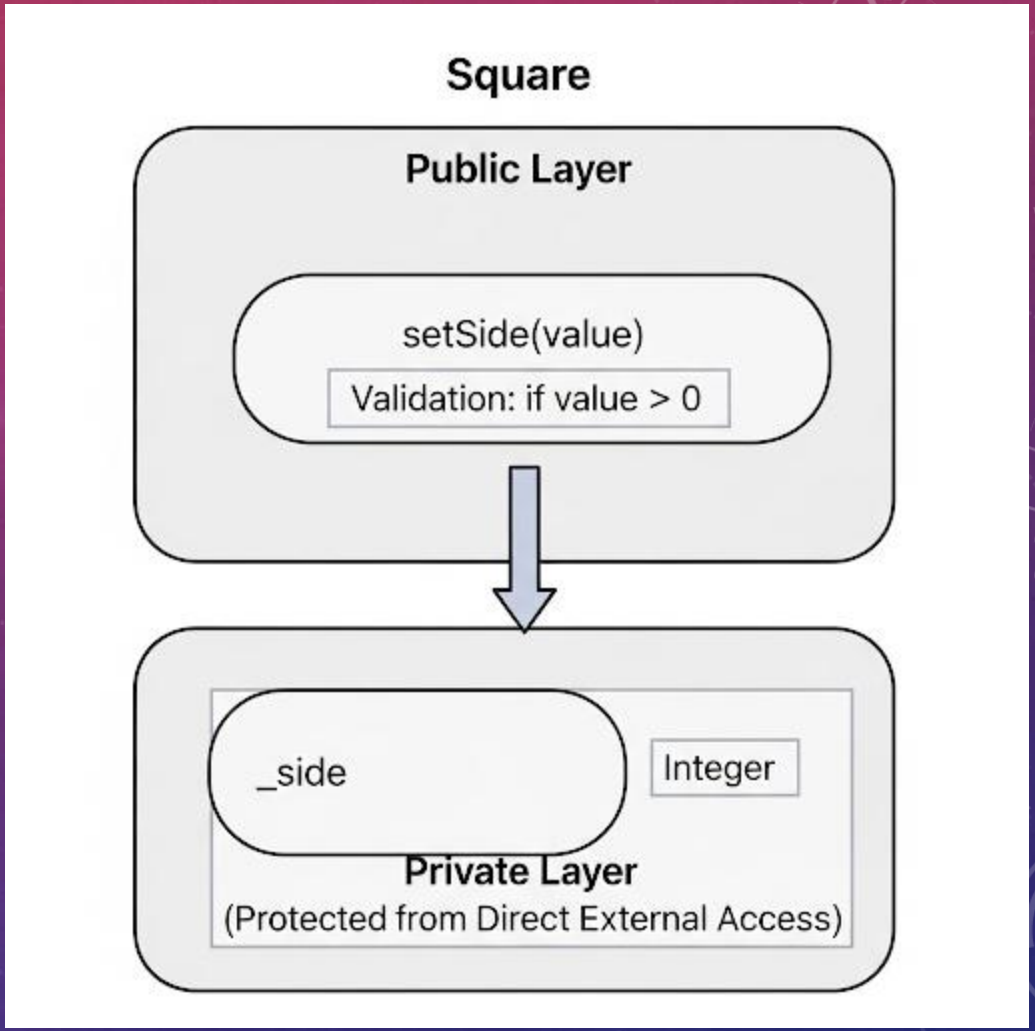
Abstract Property: An abstract property is a placeholder that forces every inherited class to provide its own specific implementation, such as the unique Area calculation for a circle or a square.

OBJECT-ORIENTED PROGRAMMING (OOP) IN C#

OOP in C# is a way to write code by thinking about things as "objects" from the real world. You create templates called "classes" to define what an object is and what it can do, and then you build many different objects from those templates.

Encapsulation

Definition: Encapsulation is the practice of bundling properties and methods into a single unit (the class) and hiding the internal details from the outside world, it's like putting a protective shell around your code.



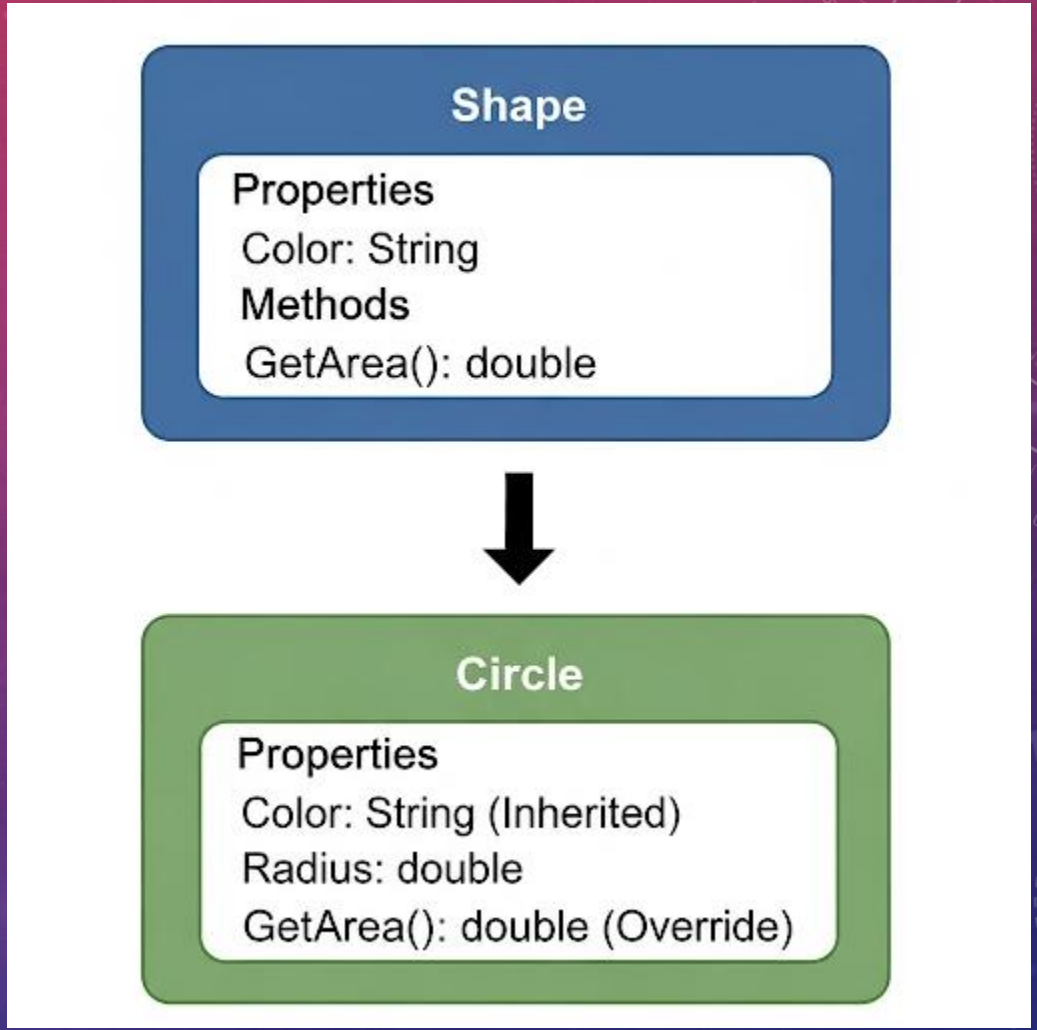
Inheritance

Definition: : Inheritance is a way where one class like sub-class can reuse and extend the properties and methods of another class main/parent.

Polymorphism

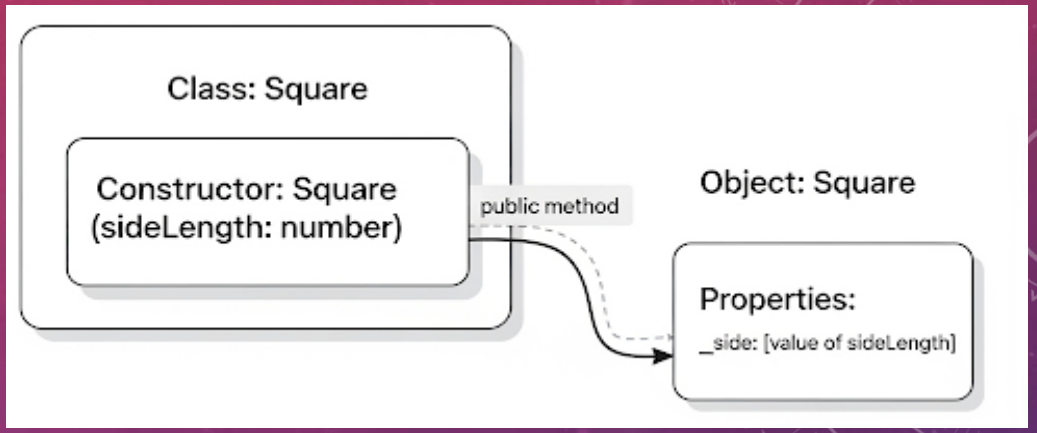
Definition: The ability for a single action to behave differently depending on the object it's performed on.

The method overriding is a form of polymorphism



Constructor

Definition: **constructor** is a special method that is automatically called when an object is created. Its job is to initialize the object's properties and ensure it starts in a valid state.




```
public class Square
{
    private double _side;

    public Square(double sideLength)
    {
        if (sideLength > 0)
        {
            _side = sideLength;
        }
        else
        {
            _side = 1;
            Console.WriteLine("Warning: Side length must be positive
        }
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Square validSquare = new Square(100);
        Square invalidSquare = new Square(-50);
    }
}
```