Inheritance

What is Inheritance?

- `Inheritance` is a concept in **OOP** where one **class** (the **child class**) can **reuse** or **extend** the properties and methods of another **class** (the **parent class**).

Why Do We Use Inheritance?

- **Reusability**:
  - Common functionality can be written once in the **parent class** and reused by **child classes**.
- **Code Organization**:
  - It helps group similar **classes** under a shared structure.
- **Extensibility**:
  - **Child classes** can add or override behavior from the **parent class** to specialize it.

How Do We Use Inheritance in Code?

- Use the `:` symbol to make one **class** inherit from another.

```
// Parent Class
public class Animal { }
// Child CLass
public class Dog : Animal { }
```

Extends

- The keyword `extends` is used when a **class** inherits from another **class**. It means the new **class** (child) will reuse or add to the functionality of the **parent class**.

SUPER

- The `super` keyword refers to the **parent class.** It is used to:
  - Call a parent class's **method**.

- Access a parent class's **constructor**.

ABSTRACT CLASS

- An `abstract class` is a class that:
  - **Cannot be instantiated** (you can't create objects from it directly).
  - Is used as a **template** for other classes to extend.
  - Can have both **concrete methods** (with code) and **abstract methods** (without code).

ABSTRACT METHOD

- An `abstract method` is a **method** with no body (just a declaration) inside an **abstract class**.
  - **Child classes** must override and provide code for abstract methods.