Inheritance

Inheritance in C# is a fundamental concept of objectoriented programming that allows ,
a new class (derived class) to inherit the properties and methods of an existing class (base class). This provides a way to reuse existing code and create more specific implementations of general concet.

Key Points

- **Base Class**: The class whose members are inherited. Think of it as the parent class.
- **Derived Class**: The class that inherits the members of the base class. Think of it as the child class.
- **Reusability**: Enables code reuse. If you have a method or property in a base class, you don’t need to rewrite it in derived classes.
- **Extensibility**: Derived classes can add new members or override inherited methods to provide specific implementations.

In C#, there are several types of inheritance that you can use to create complex and reusable code structures. Here’s a rundown:

1. Single Inheritance

- **Definition**: A class inherits from only one base class.
- **Example**:

```
public class Animal

{

   public void Eat() => Console.WriteLine("Eating");

}


public class Dog : Animal

{

   public void Bark() => Console.WriteLine("Barking");

}
```

2. Multilevel Inheritance

- **Definition**: A class inherits from a base class, and another class inherits from that derived class.
- **Example**:

```
public class Animal

{
    public void Eat() => Console.WriteLine("Eating");
}

public class Dog : Animal
{
    public void Bark() => Console.WriteLine("Barking");
}

public class Puppy : Dog
{
    public void Weep() => Console.WriteLine("Weeping");
}
```

3. Hierarchical Inheritance

- **Definition**: Multiple classes inherit from a single base class.
- **Example**:

```
public class Animal
{
    public void Eat() => Console.WriteLine("Eating");
}

public class Dog : Animal
{
    public void Bark() => Console.WriteLine("Barking");
}

public class Cat : Animal
{
    public void Meow() => Console.WriteLine("Meowing");
}
```

4. Multiple Inheritance (Interface-based)

- **Definition**: A class can implement multiple interfaces.

- **Example**:

```
public interface IWalkable

{

    void Walk();

}


public interface ISwimmable

{

    void Swim();

}


public class Duck : IWalkable, ISwimmable

{

    public void Walk() => Console.WriteLine("Walking");

    public void Swim() => Console.WriteLine("Swimming");

}
```

5. Hybrid Inheritance

- **Definition**: A combination of two or more types of inheritance.
- **Example**: C# does not directly support multiple class inheritance, but it can be achieved thro ugh interfaces.

Summary

- **Single Inheritance**: One class inherits from another.
- **Multilevel Inheritance**: Derived class is further inherited by another class.
- **Hierarchical Inheritance**: Multiple classes inherit from one base class.
- **Multiple Inheritance**: Implemented through interfaces.
- **Hybrid Inheritance**: Combination of multiple types of inheritance.