

Explanation Inheritance

Inheritance is a concept of Object-Oriented Programming in which one class (derived or child class) derive the behavior and properties of another class called base or parent-class. The child class inherits the properties and methods of the parent class, new features or functionality can be added in the child class based on your requirements.

Why inheritance is important:

Reuse of code is one of the main advantages of inheritance. The logic you share is created once in the base class and used by all derived classes, so there is less duplication of code and also easier to maintain and extend programs.

Application of inheritance:

Inheritance is typically used when the classes exhibit a hierarchy, they represent different types of activities or objects where one kind has some attributes / behaviour, and the other has something additional.

Short code example (C#):

```
public class Animal
{
    public virtual string Speak()
    {
        return "The animal makes a sound.";
    }
}

public class Dog : Animal
{
    public override string Speak()
    {
```

```
        return "The dog barks.";
    }
}
```

For instance, in the image above, Dog derives from Animal. It is a reuse of the Booth.Dog.Speak method, modified for Dog-specific behavior. This demonstrates how the inheritance lets you have shared behavior, while allowing specialization.