Object Oriented Principles

By Dhananjay Masal– CODEMIND Technology

Contact us +91 9890217463

Four pillars of OOP

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

Encapsulation (Binding)

- Process of defining a class by **hiding its internal data member direct access from outside class** & providing its access only through public exposed method(Setter & getter)
- Encapsulation hides the internal state & behavior of an objects.
- Hide data for security such as making variable private and expose property to access the private data that will be public.

Encapsulation..

How can we implemented it?

1. By declaring variables as private.
2. By defining one pair of public setter & getter method.

```
Eg. Public Class Example{

        Private int _variable;

        Public int Variable {

        get{ return _variable ;}

        set { _variable = value;}

        }

}
```

Encapsulation...

Benefit of Encapsulation:

1. Reduce dependency
2. Prevent accidental data corruption
3. Help to prevent changes to breaking your code.

Eg. When we watch TV, we need only TV & remote. To use remote key we can manage everything of TV operations.

Abstraction(Hiding)

- Process of defining a class **by providing the necessary & essential details of an object to the outside world & hiding the unnecessary things** is called Abstraction.
- I.e need to display what is necessary & compulsory and hide unnecessary things from outside.
- Abstraction let’s focus on what the object does instead of how it does.

Abstraction..

Eg. If we have below mobile phone:

1. Nokia 1400 (Feature: calling, sms)
2. Nokia 2700 (Feature: calling, sms,Radio)
3. Nokia 6600 (Feature: calling, sms,Radio, MP3, Camera)

Abstraction information (necessary & common information) for object “mobile phone” is that it makes call to any number and can send sms.

Abstraction Vs Encapsulation

1. Solves problem in **design level**
2. Used for hiding unwanted data & giving only relevant data.
3. Set focus on the object instead of how it does it
4. Outer layout in terms of design
   Eg. Outer look of IPhone, like it has display screen

1. Solves the **problem in implementation**.
2. Hiding code & data into single unit to protect data from outer world
3. Means hiding interna; details or mechanic of how an object does something.
4. Inner layout in terms of implementation
   Eg. Inner implementation details of IPhone, how display screen connect with each other using circuits.

Inheritance

- Process of creating new class from an existing class such that the new class has all properties & behaviors of the existing class.
- It is basically “is a” relationship.

  Eg. Teacher “is a” person, Student “is a” person

- All characteristic i.e. method, variables, properties(**other than private**) are inherited by child class

Inheritance

- Process of creating new class from an existing class such that the new class has all properties & behaviors of the existing class.
- It is basically “is a” relationship.

  Eg. Teacher “is a” person, Student “is a” person

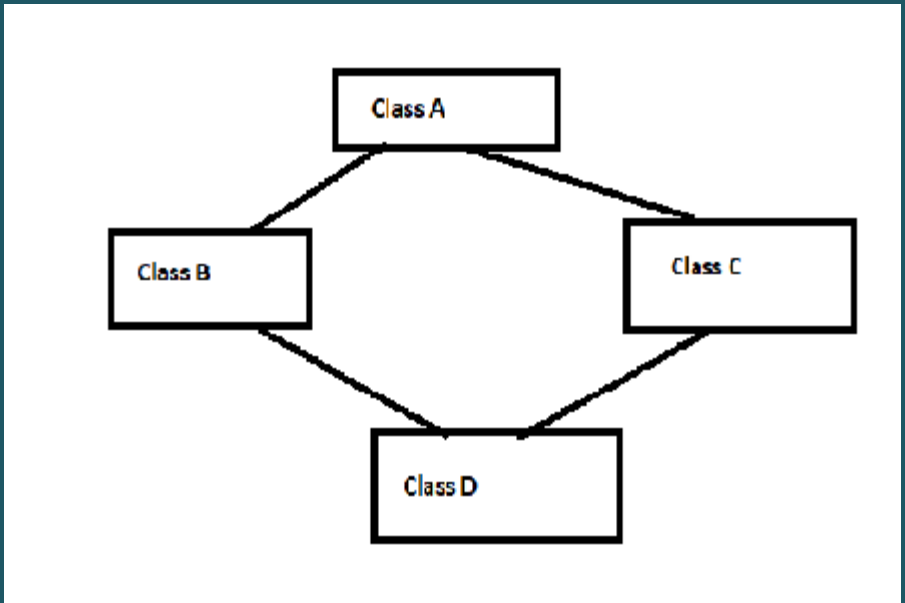
- All characteristic i.e. method, variables, properties(other than private) are inherited by child class

Inheritance..

1. C# support only single class inheritance.
2. C# Supports multiple interface inheritance, not class multiple inheritance.

Why C# does not support multiple Inheritance?

Diamond problem:



Diamond Problem..

- If there is a method in Class A that Class B and Class C have overridden & Class D does not overridden it, then which class of method Class D inherit.
- Ambiguity problem in multiple inheritance.

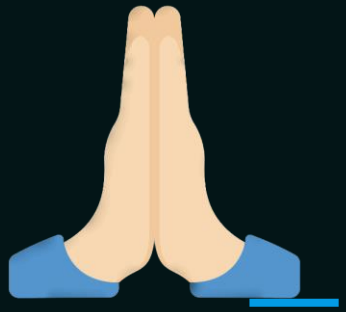
Note: Multiple inheritance is possible in C# while we will use atleast one base class as interface.

Types of inheritance

1. Single inheritance: Class derived from single class
2. Multilevel inheritance: Derived class created from another derived class.
3. Hierarchical inheritance: More than one derived class is created from single base class
4. Hybrid inheritance: Combination of any single, Hierarchical, Multilevel
5. Multiple inheritance: Class derived from multiple class. Not supported in C#

```
Public class Person{
string Name;
String Address;
}
Public Class Student : Person {
Int RollNo;
}
```

Thank You



Success is not a milestone, it's a journey. And we have vowed to help you in yours.

www.codemindtechnology.com