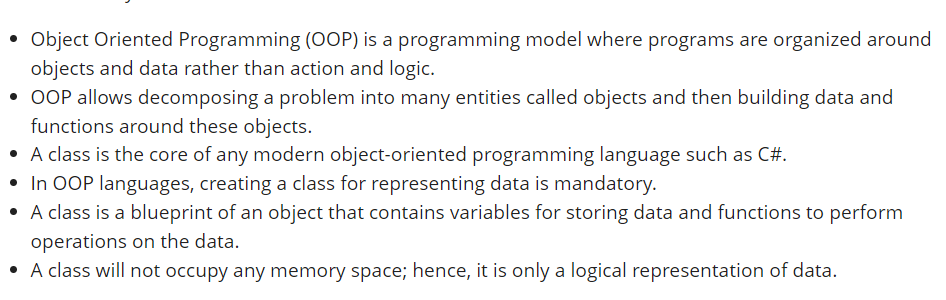
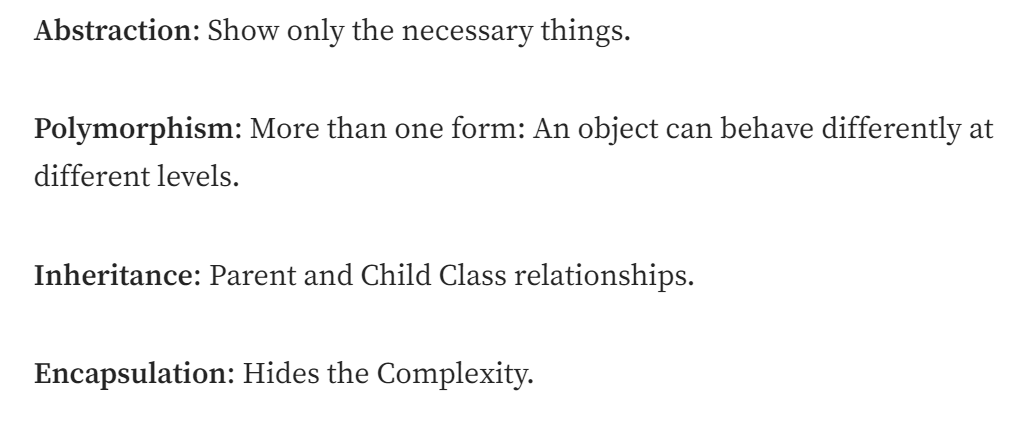
Object Oriented Programming Concepts in C#



## **OOPs Concepts**

The key concepts of OOPs are

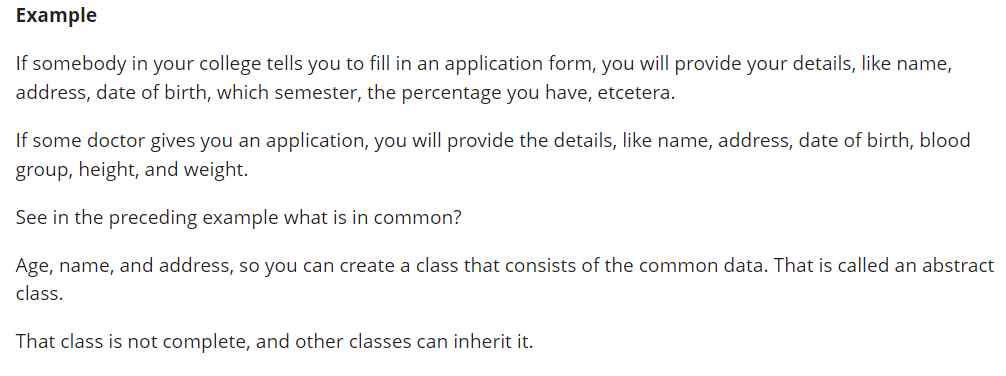
1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism



## **Abstraction**

Abstraction is "To represent the essential feature without representing the background details."

* Abstraction lets you focus on what the object does instead of how it does it.
* Abstraction provides a generalized view of your classes or objects by providing relevant information.
* Abstraction is the process of hiding the working style of an object and showing the information about an object understandably.



## **Encapsulation**

Wrapping up a data member and a method together into a single unit (in other words, class) is called Encapsulation. Encapsulation is like enclosing in a capsule. That is, enclosing the related operations and data related to an object into that object.

* Encapsulation means hiding the internal details of an object, in other words, how an object does something.
* Encapsulation prevents clients from seeing its inside view, where the behavior of the abstraction is implemented.
* Encapsulation is a technique used to protect the information in an object from another object.
* Hide the data for security, such as making the variables private, and expose the property to access the private data that will be public.

In OOP, encapsulation is typically achieved through the use of access modifiers, such as “private” and “protected,” which restrict access to certain members of a class.

## **Inheritance**

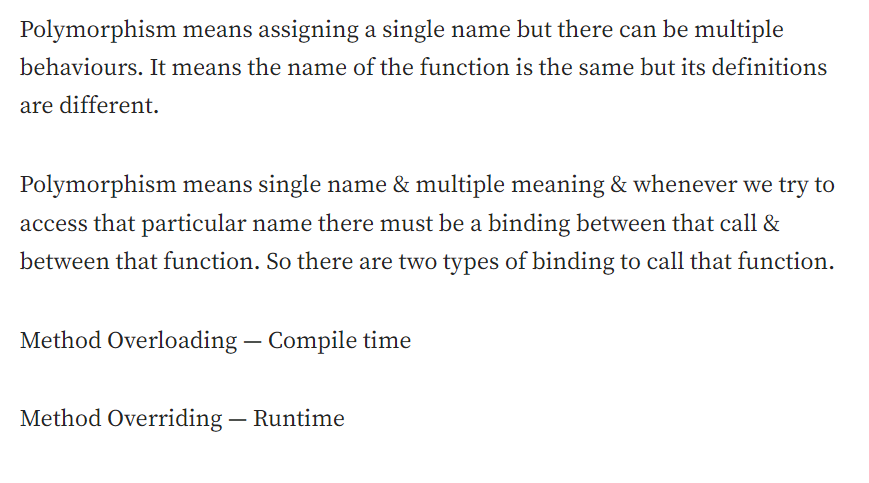
When a class includes a property of another class, it is known as inheritance. Inheritance is a process of object reusability.

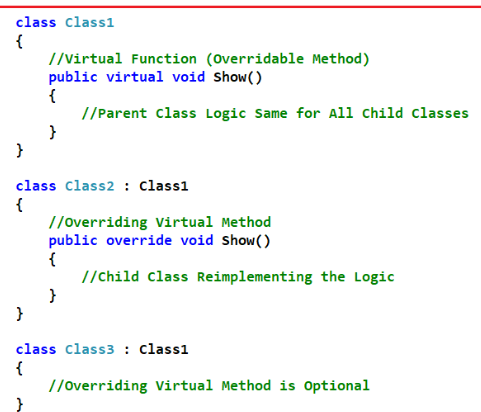
Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and speeds up implementation time.

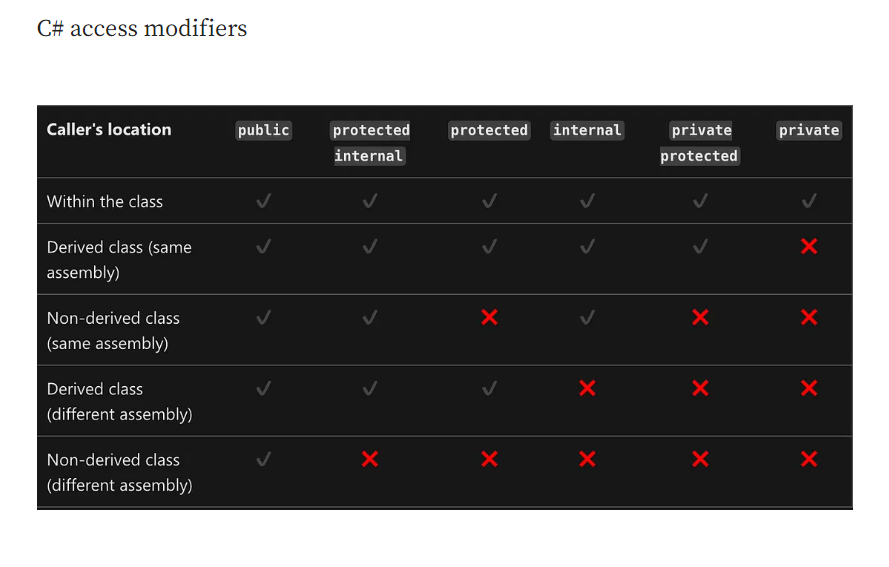
When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base** class, and the new class is referred to as the **derived** class.

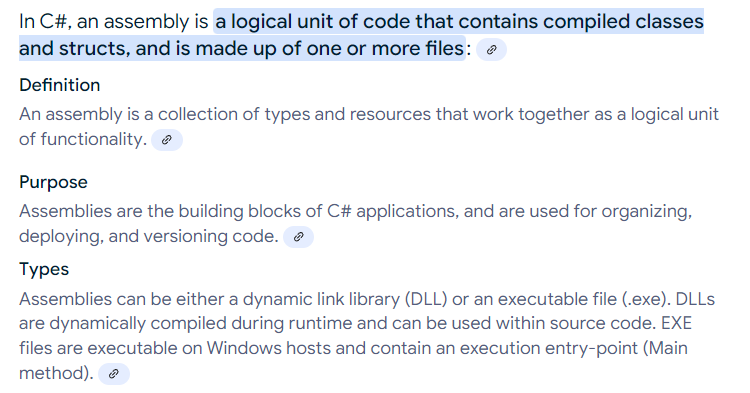
## **Polymorphism**

Polymorphism means one name, many forms. One function behaves in different forms. In other words, "Many forms of a single object is called Polymorphism."









**OOP Theory**

**Why do we need OOP ?**

OOP helps us to think in terms of real world object.

Eg: Class patient { public string name; public doctor doct;}

Class doctor{ public string name; }

**What are important pillars of OOP ?**

Abstraction : Show only what is necessary.

Polymorphism: Object acts differently under different conditions.

Eg: User object can behave as Employee, Admin , worker etc..

Inheritance : Parent child relation

Encapsulation: Hide Complexity

**What is a class and Object ?**

Class is a blue print or a type and object is instance of class.

Abstraction vs Encapsulation ?

Abstraction happens during design phase.

Eg: While designing a class, developer decide what has to be public or private method or properties using access modifiers inorder to hide complexity.

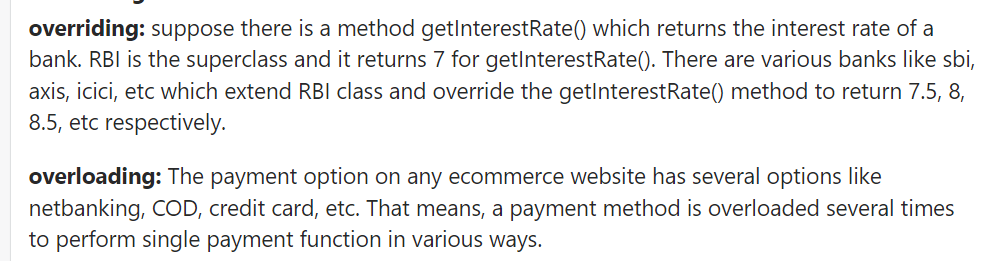
Encapsulation happens in execution phase. It implements abstraction.

Actually both compliment each other.

Override and Virtual keyword ?

Virtual keyword helps us to define some logic in parent class which can be overridden in the child class. Parent child relationship is required

Real time examples of Override and Overload.



Polymorphism

Ability of an object to act differently under different condition

Eg:

Employee e = new Employee();

e = new Supervisor();

Can polymorphism work with out inheritance ?

No

Static vs Dynamic Polymorphism

Static Poly – Method overloading

Dynamic Poly – Method Overiding

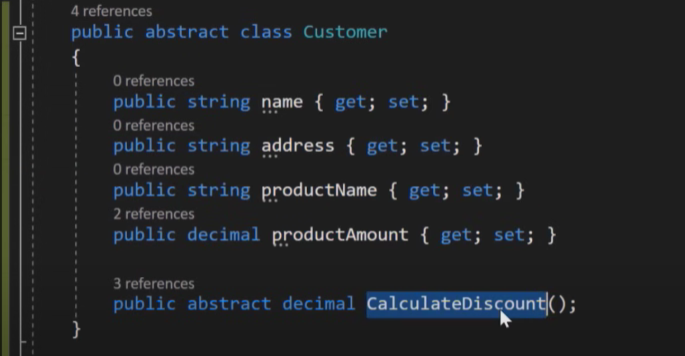
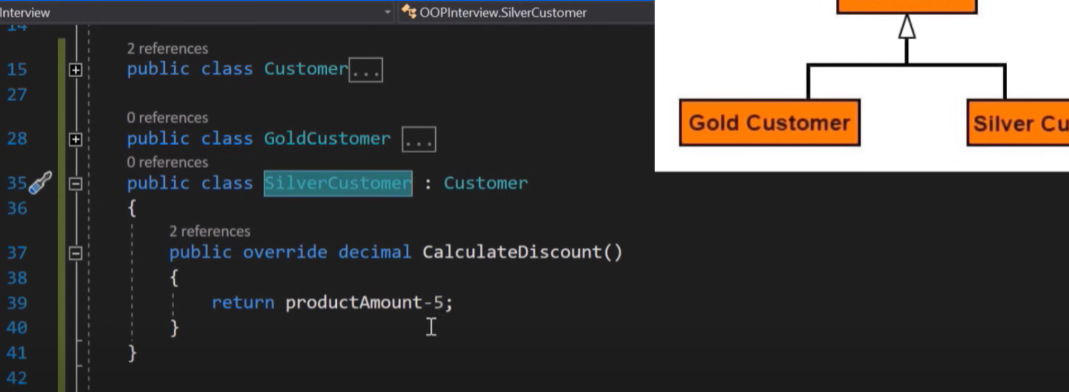
Operator Overloading

It helps to redefine additional functionalities for plus,minus multi and addition.

Why do we need Abstract classes ?

It is a half defined parent class or partially defined parent class.

Why Simple base class replace Abstract class ?

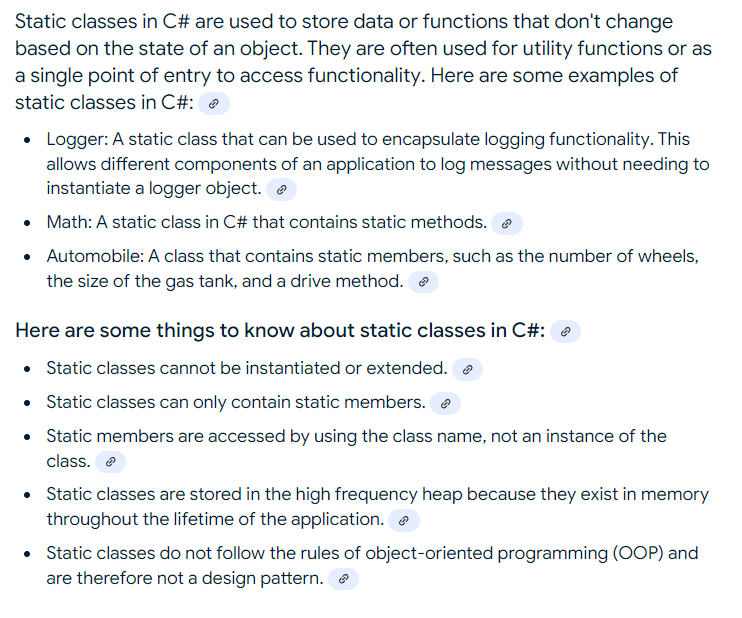
Abstract methods in abstract class are by default virtual.

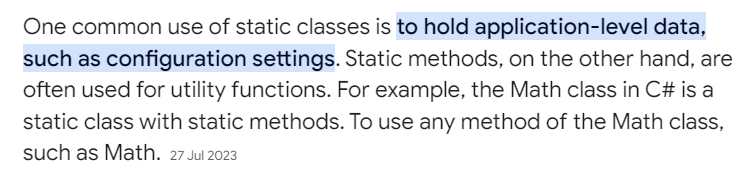
What is Static class & When to use ?

It is a class which object cannot be created and which cannot be inherited.

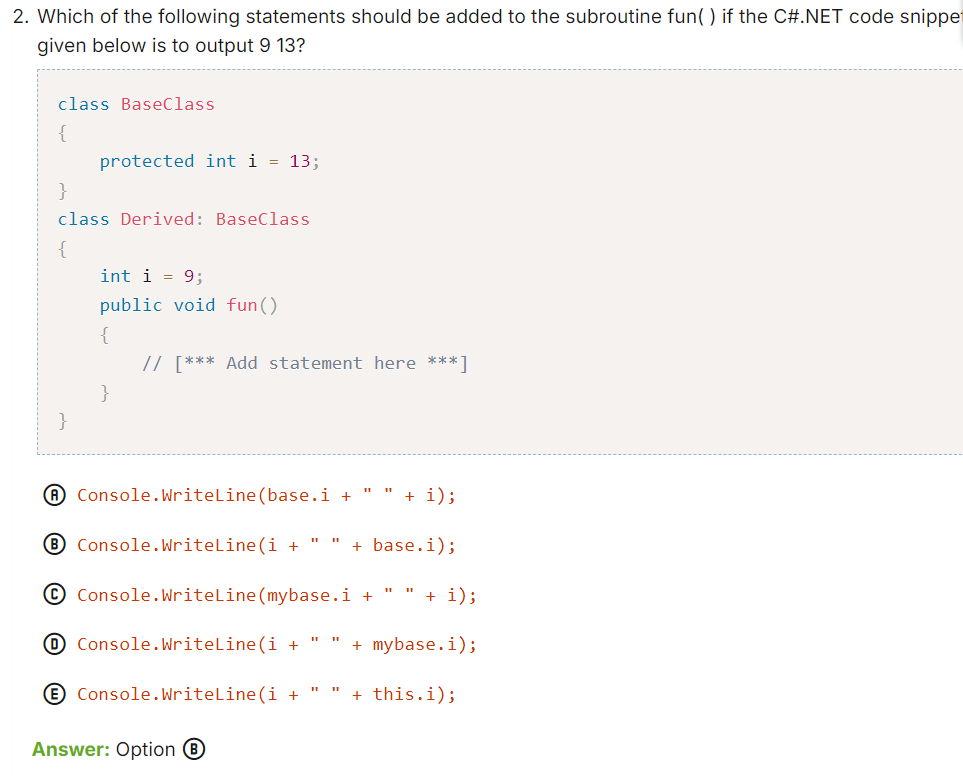
Static classes are used as containers for static members like methods, constructors and others.

Every members of static class should be static.



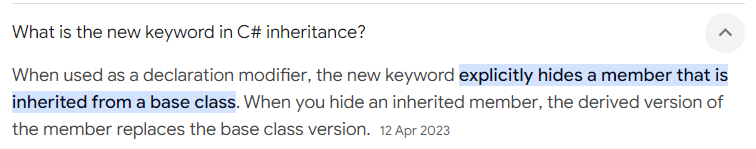


Programming Questions:



In BaseClass 'i' is local variable, and inside DerivedClass 'i' is again a local variable of it, So, 9 value scope is inside the DerivedClass and 13 scope is outside the DerivedClass, so, to call the base class members we should use "base" keyword console.WriteLine(i+""+base.i);.

**To access parent class values in derived class, use ‘base’ keyword.**



Var vs Dynamic

A screenshot of a chat

AI-generated content may be incorrect.