**Single Responsibility:**

1. A class should take one responsibility and there should be one reason to change that class.
2. It does not mean that your classes should only contain one method or property. There may be many members as long as they relate to single responsibility.

**Open close principle:**

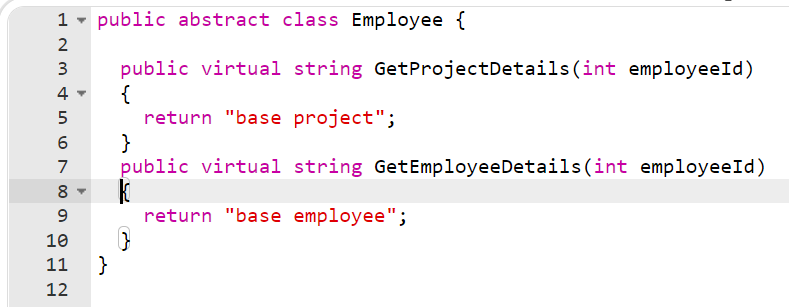
**Definition:** Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification.

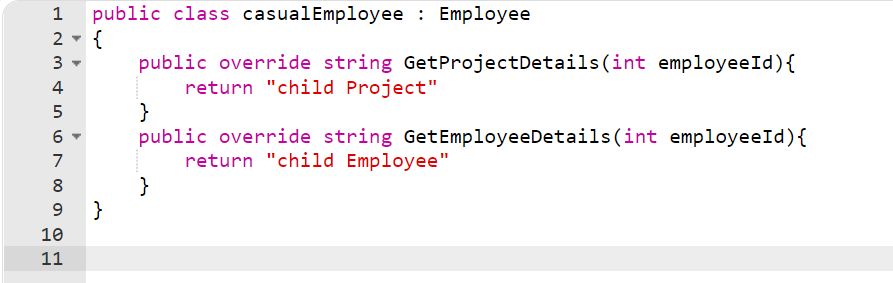
**Understanding:**

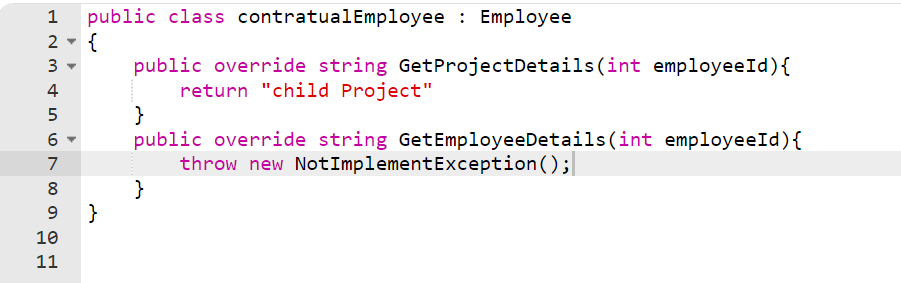
This principle suggests that the class should be easily extended but there is no need to change its core implementations.

**i.e.** New features should be implemented using the new code, but not by changing existing code. The main benefit of adhering to OCP is that it potentially streamlines code maintenance and reduces the risk of breaking the existing implementation.

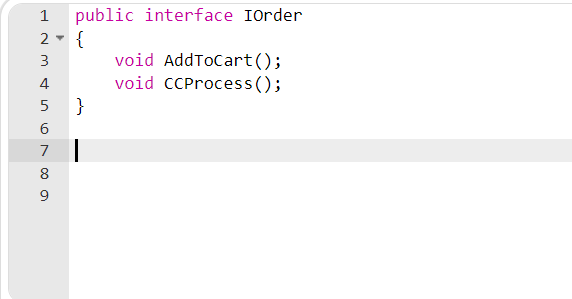
**Liskov substitution principle (LSP):**

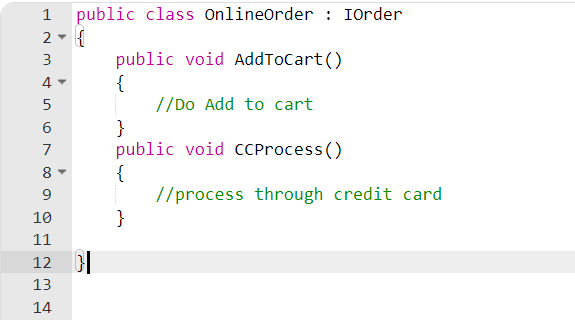
1. Child class should not break the parent class's type definition and behavior.

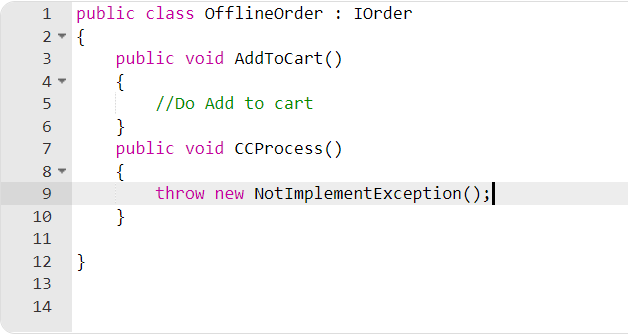




**Interface Segregation Principle (ISP):**

**Definition:** No client should be forced to implement methods which it does not use, and the contracts should be broken down to thin ones.





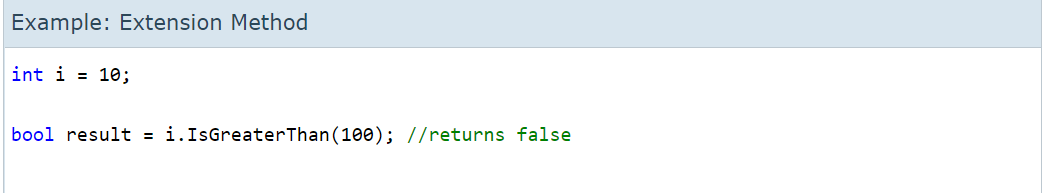
**Dependency Inversion Principle (DIP):**

* High-level modules should not depend on low-level modules. Both should depend on abstractions.
* Abstractions should not depend on details. Details should depend on abstractions.

**What is middleware?**  
A middleware a component that is executed on EVERY REQUEST in the ASP.NET Core application

**What is the extension method?**

Extension methods, as the name suggests, are additional methods. Extension methods allow you to inject additional methods without modifying, deriving or recompiling the original class, struct or interface. Extension methods can be added to your own custom class, .NET framework classes, or third party classes or interfaces.



**Abstraction:**

Abstraction means showing only required things and hiding the background details.

The abstract keyword is used for classes and methods:

* Abstract class: is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).
* Abstract method: can only be used in an abstract class, and it does not have a body. The body is provided by the derived class (inherited from).

## What Does Polymorphism Mean?

Polymorphism, in C#, is the ability of objects of different types to provide a unique interface for different implementations of methods. It is usually used in the context of late binding, where the behavior of an object to respond to a call to its method members is determined based on object type at run time.

Overloading:

1. Method overloading is a type of polymorphism, in which we can create multiple methods of the same name in the same class, and all methods work in different ways.

Overriding:

1.Method overriding is having methods with the SAME NAME and SIGNATURE but in different classes.