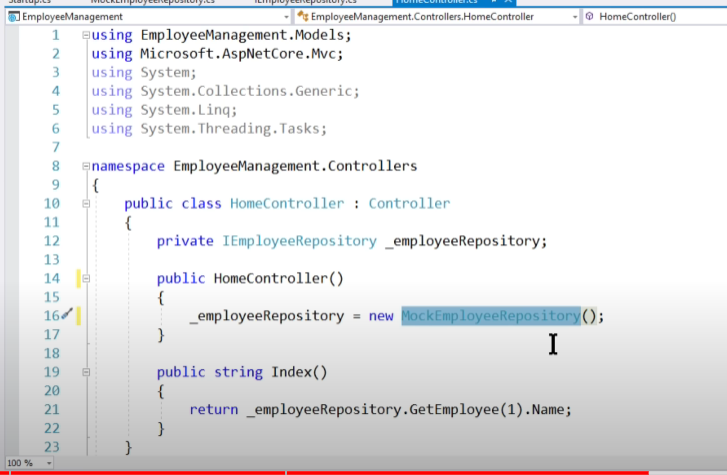
**Dependency Injection in ASP.NET CORE C#**

Traditional approach issue:

Here, we create link interface and implemented class object with new keyword in constructor.

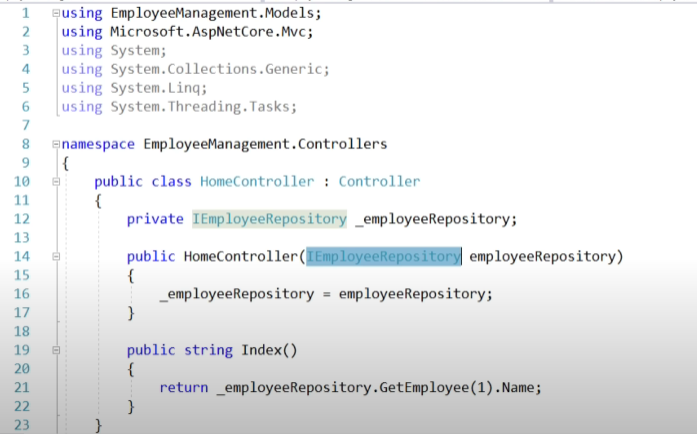
Now it’s tightly coupled with HomeController & MockEmployeeRepo class. Later if we provide an implementation of IEmployeeRepo and we need to use new implementation in HomeController then the code in controller needs to be changed

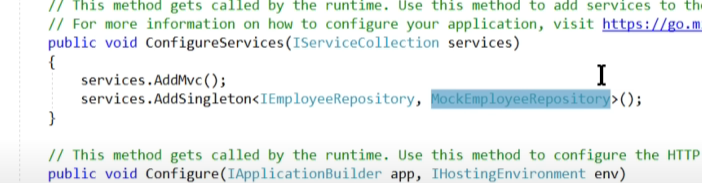
Eg: Instead of MockEmployeeRepo class, we may use new class EmployeeSQLRepo which inherits IEmployeeRepo interface. Then here controller also needs to be changed which makes it difficult to handle and tds work to change in all places in case of larger project.



In Dependency injection we can avoid this issue, since only interface is injected in constructor level and the configuration of Interface and class in made in startup.cs.

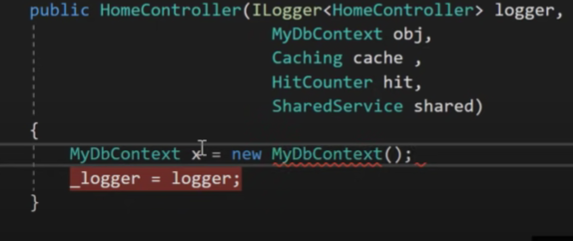
So even if a new implementation is made eg: EmployeeSQLRepo, only change is required in config level at startup.cs and HomeController code remains same as below:





A dependency injection is a methodology where in rather than caller creating the instances, it is injected by someone else.

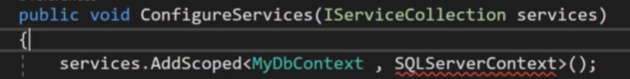
Wrong :



Benefit of Dependency Injection : Decoupled architecture, where you change at one place it can reflect at many places

In above example, DBContext is injectected in HomeController, he has no idea from where it is coming.

We change in implementation of DBContext at configure services and it can reflect in all places.



Scoped vs Transient vs Singleton : all three defines the lifetime of the dependency injected object.

Singleton : Throughout the server there is only one instance is created. (across multiple users/request) Eg: Caching object, shared data (hit counter), loggers, email services etc..

Scoped : For every request new instance will be injected. In the same request if we do dependency injection again multiple times, same instance is injected. (within the same request, the object will be same if we asking for multiple times.)

Eg: Repository for same transaction, business objects , used most of the time.

Transient : For every request new instance will be injected, but within the same request, if we do multiple DI multiple times, different instances will be created, (within the same request, the object will be created new if we are asking for multiple times)

Eg: When you want separate copies of objects, where objects are not connected each other. (a different transaction) . MultiThreading

**Inversion of Control** :

**C# Interval Questions**

**Q.** Can a static class contains non-static members?

No, a static class cannot contain non-static members. Static classes can only contain static members.

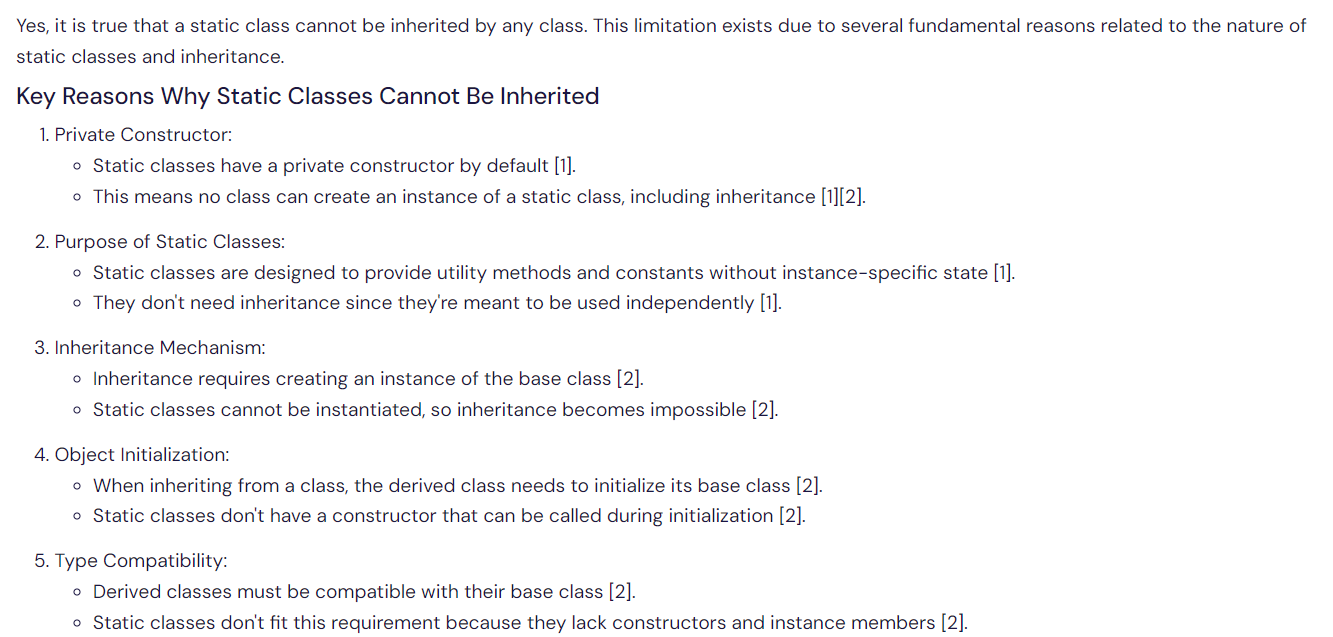
**Q.** . Can a static class inherit from another class?

No, a static class cannot inherit from another class. Static classes have limited inheritance capabilities compared to regular classes.

Q. Can a static class implement the interfaces?

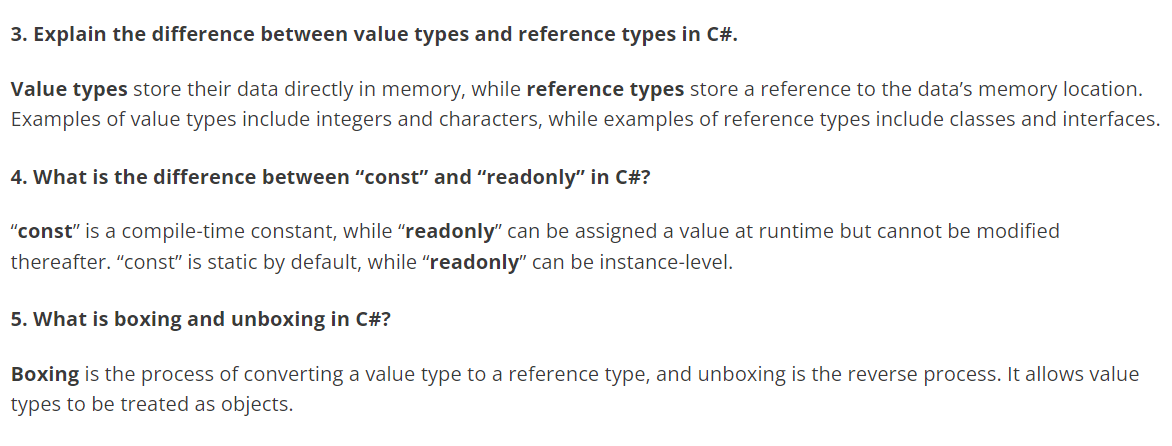
No, a static class cannot implement interfaces. Static classes have limited functionality compared to regular classes.

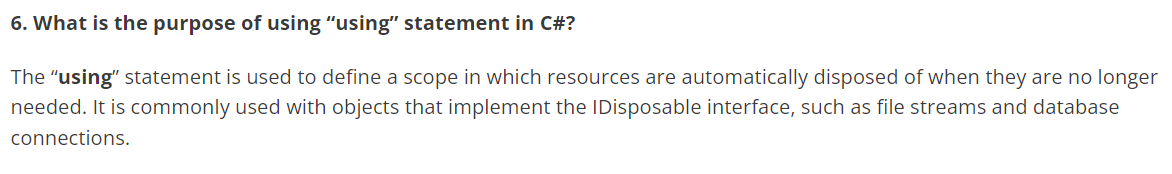
Q. Is it true that a static class cannot be inherited by any class, why?

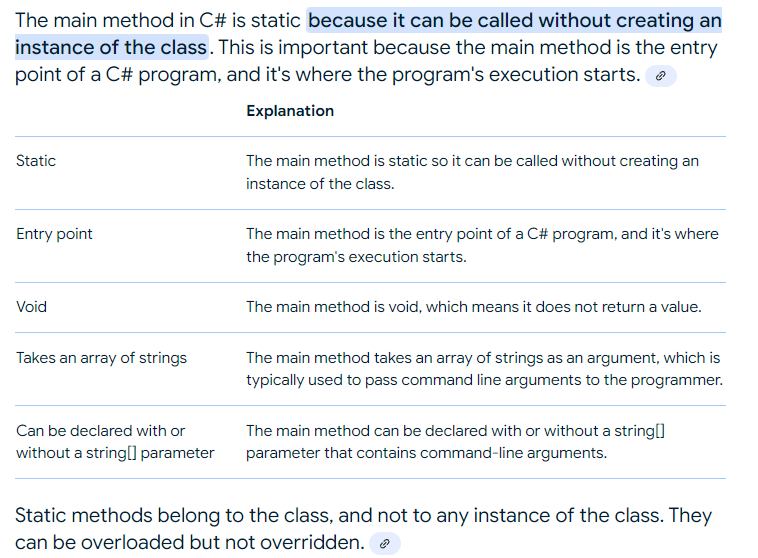


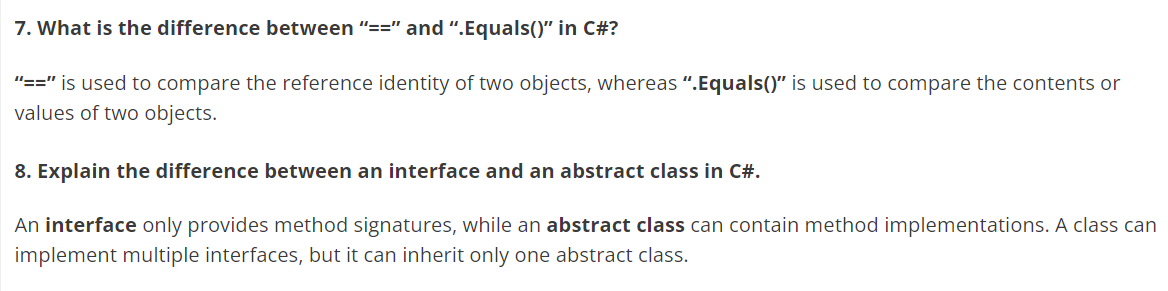
 Q. Where do static classes and members are stored in the memory?

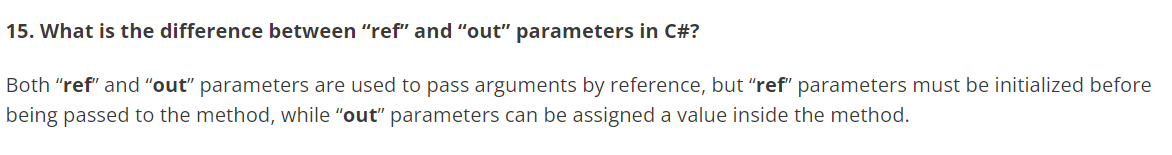
Static objects are stored in the high frequency heap. The CLR divides system memory into three regions: the stack, the heap, and the high frequency heap.



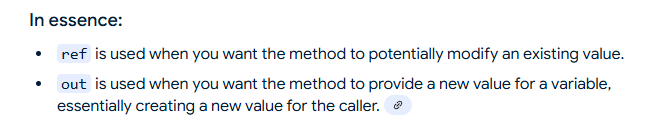








Especially used to return multiple value from a method.

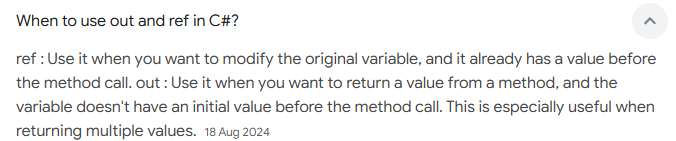
****

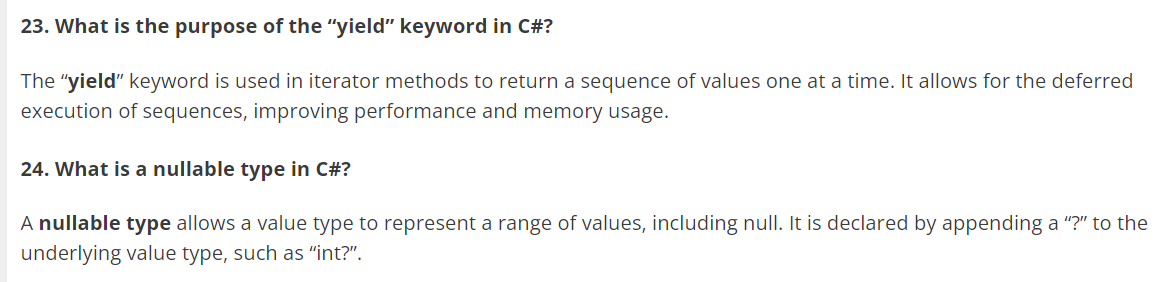
****

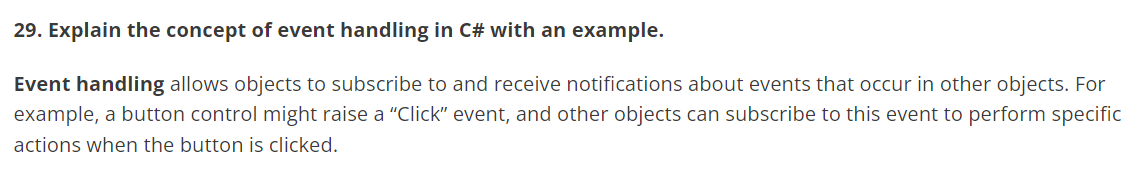
**Difference:**

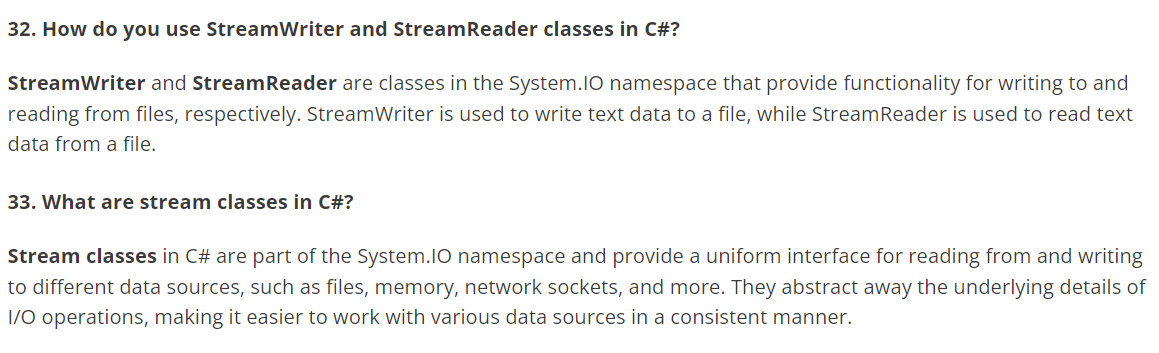
**I**f you are declaring some out variables, then it is mandatory or compulsory to initialize or update the out variables inside the method body else we will get a compiler error. But with the ref, updating the ref variable inside a method is optional.

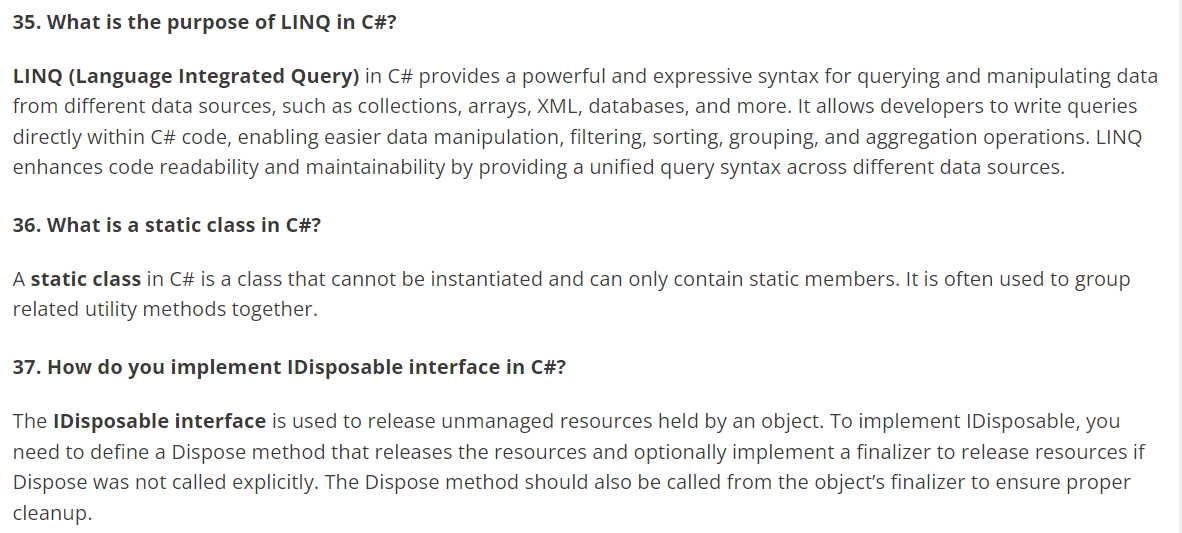
Initializing the ref parameter is mandatory before passing such variables to the method while initializing the out-parameter variables is optional in C#

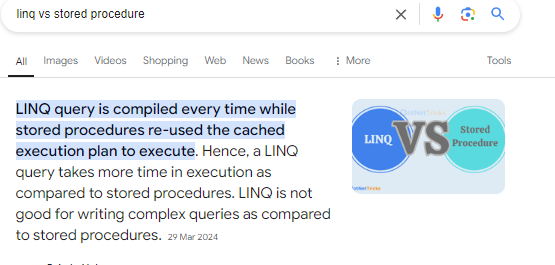


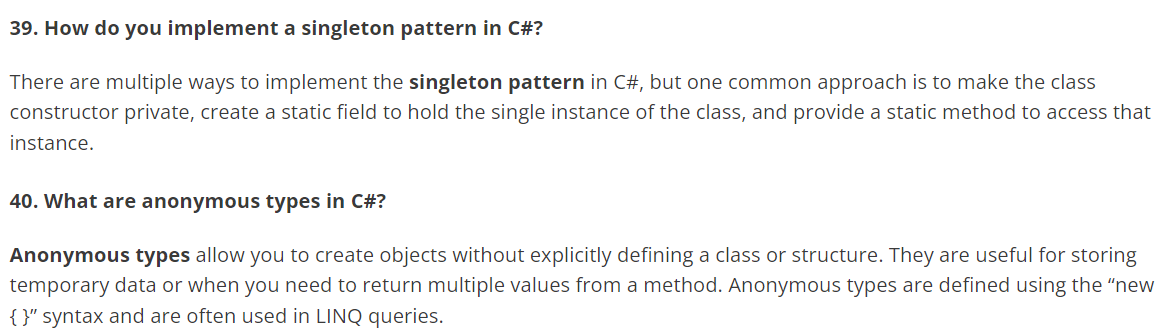








****



**IQuerable & IEnumerable**

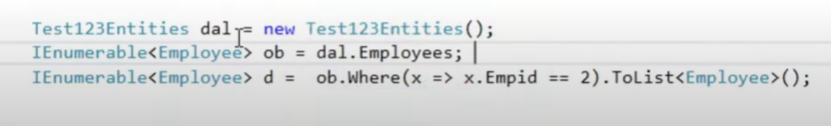
Both are interfaces to .NET collection.

IQuerable inherits from IEnumerable, what things IEnumerable can do IQuerable can also do.

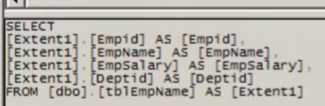
But whatever things IQuerable can do, IEnumerable cannot do.

**Public class IQuerable : IEnumerable**

IEnumerable -

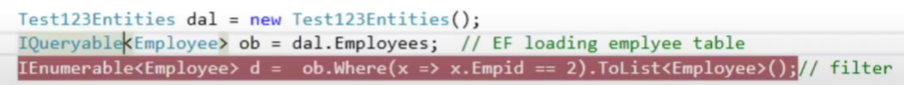
****

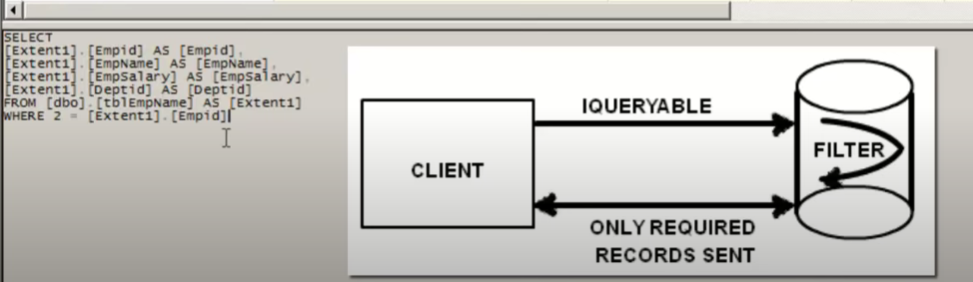
Here, it only executed the select statement from DB and loads all data to memory.

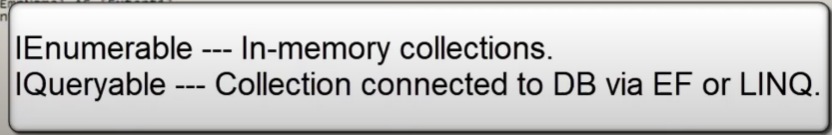


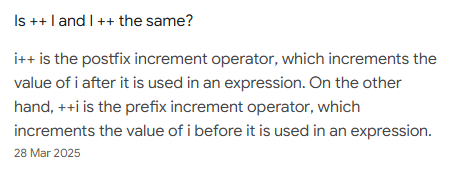
And the where condition will be applied to memory stored data.

IQuerable – Filter condition is directly applied on DB and returns only required data.

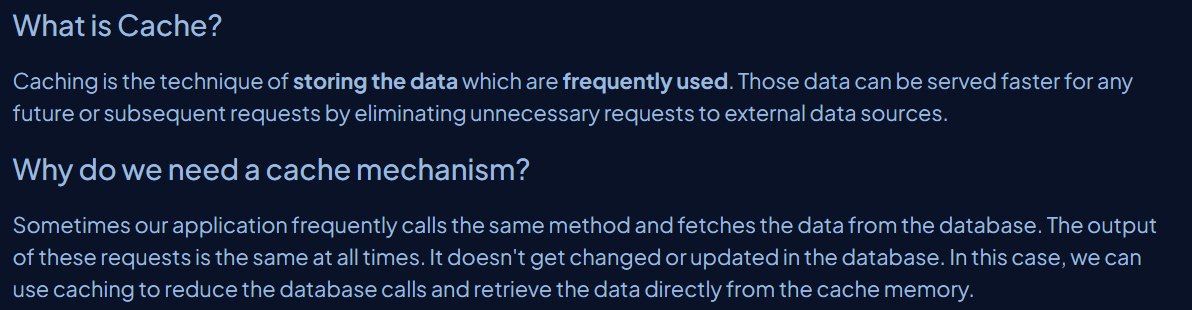
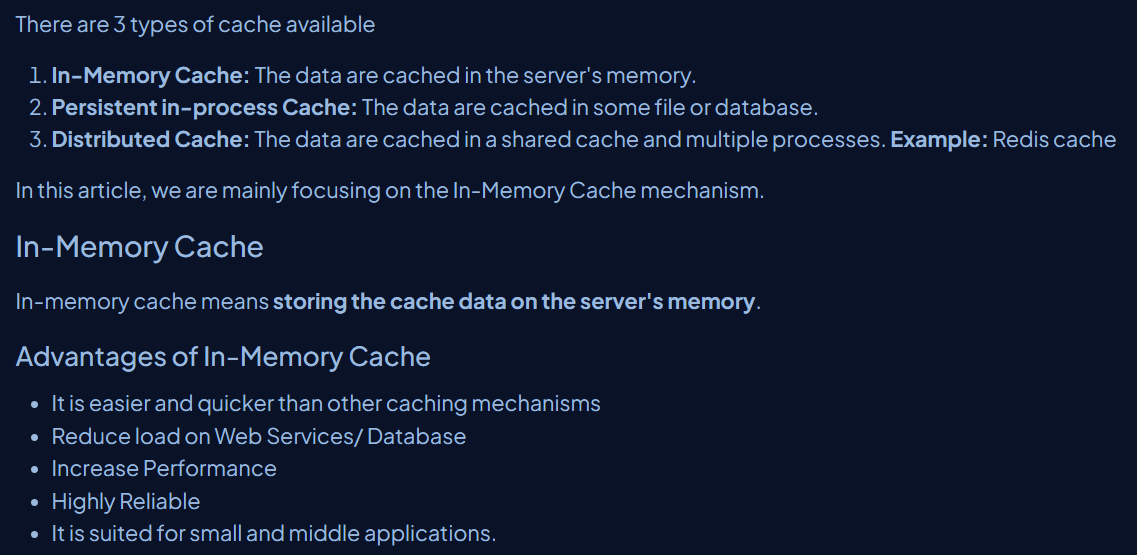


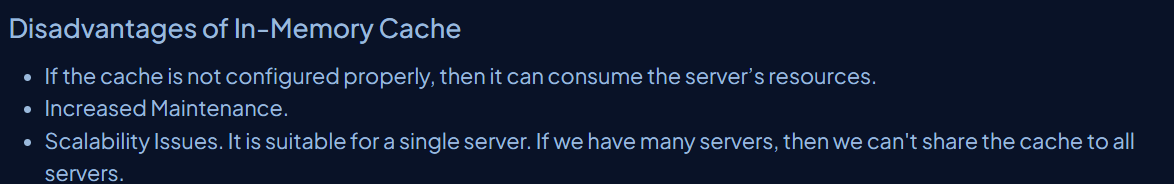


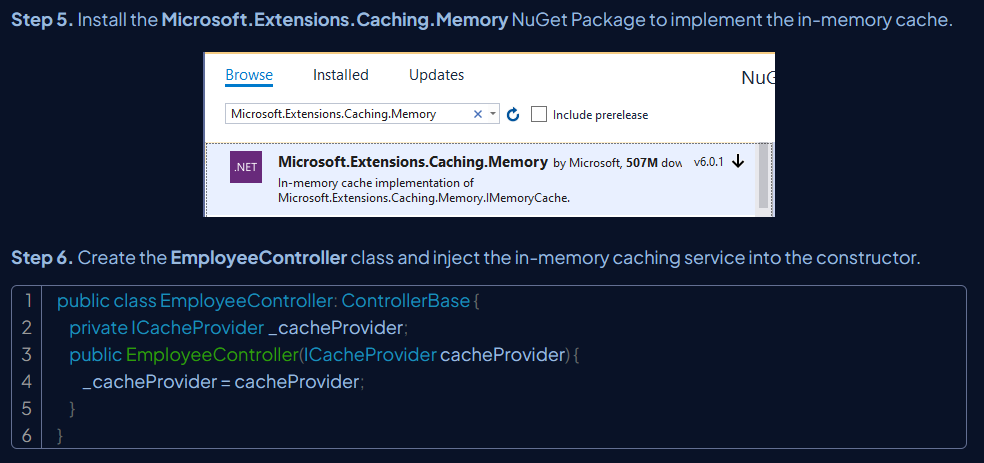


****

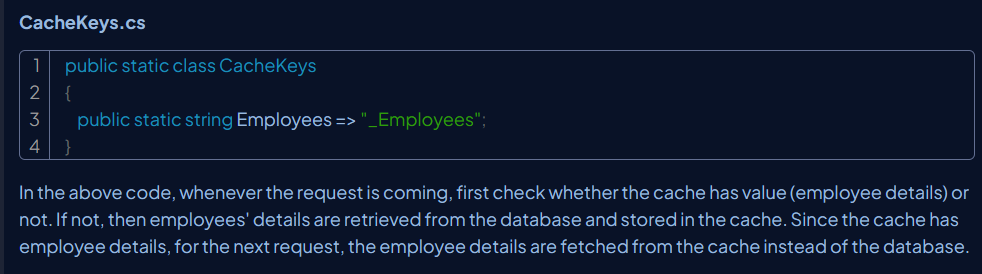
**Caching Implementation in .NET Core**

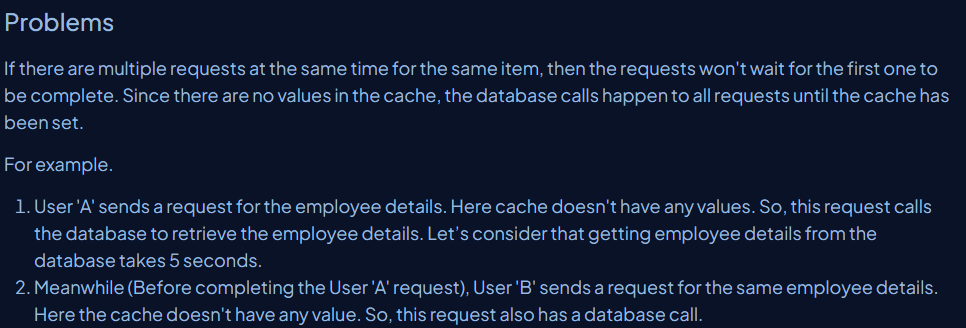
** **

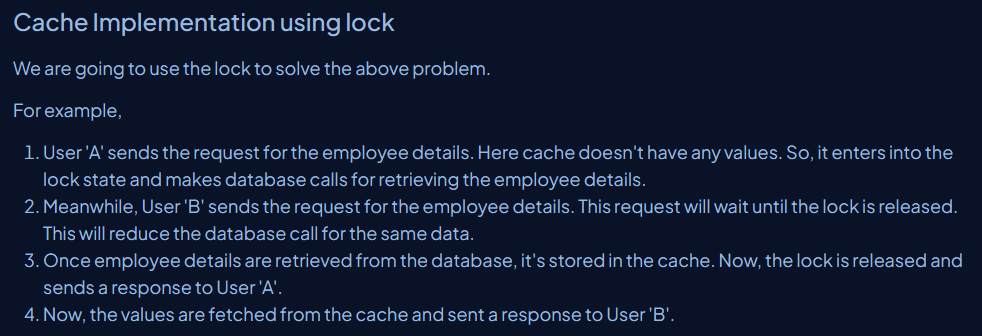
****

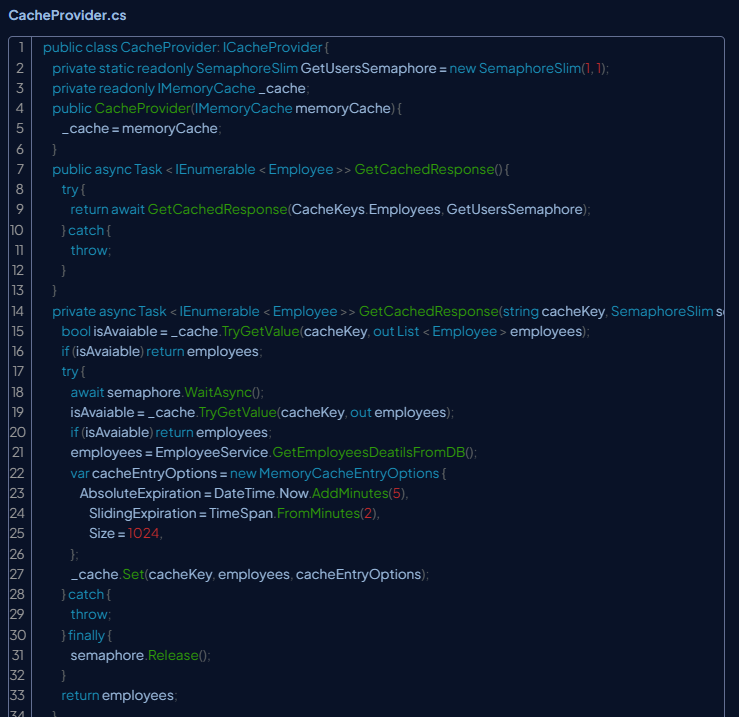
****

****

****

****

****

****

**Design Patterns in C#**

A close-up of a black and white background

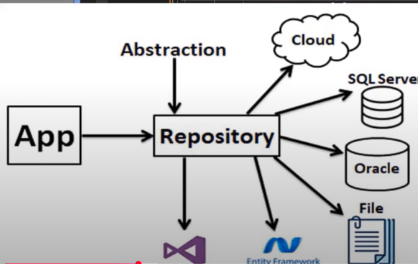
AI-generated content may be incorrect.

A close up of a card

AI-generated content may be incorrect.

A close up of a sign

AI-generated content may be incorrect.



A close up of a sign

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.



A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with text and images

AI-generated content may be incorrect.

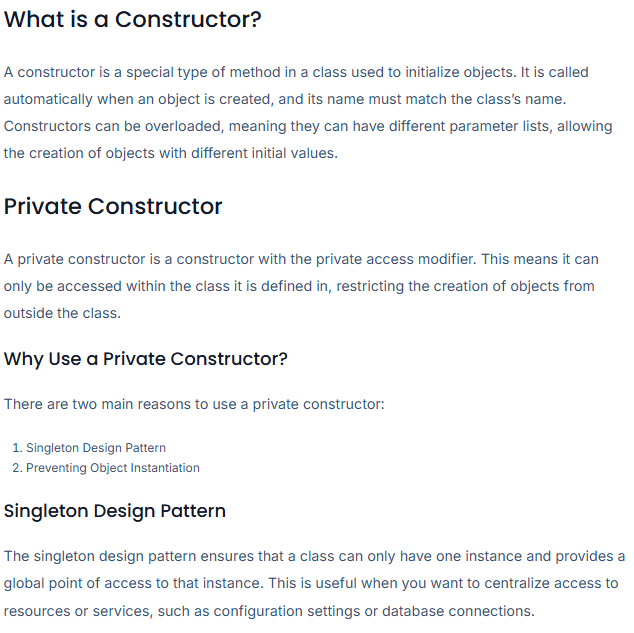
SingleTon Pattern

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.



A screenshot of a web page

AI-generated content may be incorrect.

A white text on a white background

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.

**C#**

A screenshot of a computer

AI-generated content may be incorrect.

**Exception handing in c# ?**

**Throw vs throw ex**

The difference between “throw” and “throw ex” is that “throw” preserves the stack trace (the stack trace will point to the method that caused the exception in the first place) while ”throw ex” does not preserve the stack trace (we will lose the information about the method that caused the exception in the first place.

A screenshot of a computer program

AI-generated content may be incorrect.

**Program.cs ?**

Program.cs is where the application starts. Program.cs file will work the same as the Program.cs file in traditional console application of .NET Framework. Program.cs file is the entry point of the application and will be responsible for registering Startup.cs fill, IISIntegration and Create a host using an instance of IWebHostBuilder, the Main method.

A screenshot of a computer

AI-generated content may be incorrect.

**StartUp.cs ?**

Global.asax is no longer in the ASP.NET Core application. The Startup.cs file is a replacement of the Global.asax file in ASP.NET Core.

Startup.cs file is entry point, and it will be called after Program.cs file is executed at the application level. It handles the request pipeline. The startup class triggers the second the application launches.

We can define two methods in the startup file ConfigureServices and Configure along with the constructor.

A screenshot of a computer program

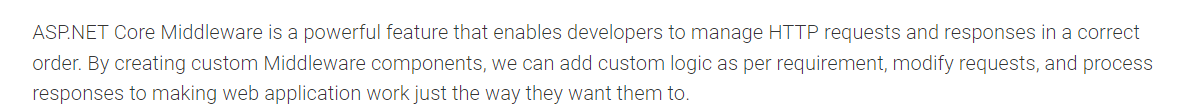
AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.



A close-up of a computer screen

AI-generated content may be incorrect.

### **Can multiple catch blocks be implemented?**

Multiple catch blocks cannot be implemented. Once you execute the proper catch code, the control is moved to the final block. After this, the code following the final block is implemented.

**Logs in C#**

### **Which one is better/faster, switch or if-else-if statements, and why?**

Among these two, the switch statement is considered faster than the if-else-if statement. This is because the switch does not check earlier statements, but in case of if-else-if each condition has to be checked

### **When should you use Async and Await?**

### Async and Await are used in C# to create asynchronous methods. Asynchronous programming processes can run without any dependency on other processes including main processes.

**Is it possible to store mixed data types such as string, int, float, and char in one array?**

**A screenshot of a computer

AI-generated content may be incorrect.**

**What do you understand about boxing and unboxing?**

**Design Patterns ?**

**Abstract vs Interface ? and use**

**What is dependency injection ?**

**It is a software design pattern that allows to develop loosely coupled application.**

**This is a process in which we are injecting the object of a class into another class that depends on that object.**

**Wrong Approach :**

A computer screen shot of a code

AI-generated content may be incorrect.

**What is loosely coupled application ?**

**Nowadays, each enterprise wants their application to be loosely coupled.**

A diagram of a diagram of bubbles

AI-generated content may be incorrect.

**In loose coupling, if any changes are made in module 1, then the other modules not need to be tested. They are using each other but not dependent.**

**On otherhand, all modules are dependent on each other, and changes in 1 module can impact other modules and all modules need to be tested.**

ConfigureAwait(true): Runs the rest of the code on the same thread the code before the await was run on. Not necessarily the same thread, but the same synchronization context. The synchronization context can decide how to run the code. In a UI application, it will be the same thread.

Use ConfigureAwait(false) when you don't need to access the original context or its state in the continuation. For example, if you are doing some CPU-bound or I/O-bound work that doesn't depend on any specific context.

‘

### Solid Principles in C#

### A screenshot of a computer AI-generated content may be incorrect.

### A computer screen shot of a computer AI-generated content may be incorrect.

### Eg: One Class should do only one thing at a time. If it is doing multiple things break the class.

### A computer screen shot of a program AI-generated content may be incorrect.

### Here is CalculateSalary is an odd one and doesn’t needs to be in Employee class.Soln : Create a new class for calculating the salary

### A computer screen with text on it AI-generated content may be incorrect.

### A close-up of a toy AI-generated content may be incorrect.

### The class should be open for extension and closed for modification.

### Eg: Here, for every employee type the salary is being calculated:

### 

### Later, if a new employeeType is introduced then also the class/code needs to be changed, which is wrong. This SalaryCalculation is used in 100 of places if any changes mage, then the whole scenarios needs to be tested.

### Soln : Rather than modifying the class, go and use inheritance.

### A computer screen with text on it AI-generated content may be incorrect.

### A duck and a duck toy AI-generated content may be incorrect.

### The Child class should be able to implement all the method of the parent class seamlessly and smoothly.

### Eg : Employee is the base class and it is inherited by InternalEmployee & ContractEmployee class.

### 

### Here ContractEmployee doesn’t have healthInsurance, so inheriting from Employee class will cause a issue since HealthInsurance is available in Employee.

### Soln : A base Employee should be added on top of all and he will have everything else HealthInsurance which can be inherited by other employee classes.

### A computer screen with text AI-generated content may be incorrect.

### A screen shot of a computer AI-generated content may be incorrect.

### A close-up of a computer device AI-generated content may be incorrect.

### Client should not to be forced to use Methods which it does not need.

### Segregate the interface.

### A screen shot of a computer program AI-generated content may be incorrect.

### In HomeController, we need to read, update & delete the data, so we can use ICrud here

### A computer screen with text AI-generated content may be incorrect.

### But in Report Controller shows on data, and it doesn’t need any other CRUD operation other than Read.

### Soln : As per principle, go ahead create a new interface as IRead, which can be used only for ReportController and ICrud will be inherited from IRead

### A screen shot of a computer AI-generated content may be incorrect.

### A close-up of a computer AI-generated content may be incorrect.

### 

### The ref keyword passes arguments by reference. It means any changes made to this argument in the method will be reflected in that variable when the control returns to the calling method.

### The out keyword passes arguments by reference. This is very similar to the ref keyword.

### A screenshot of a computer AI-generated content may be incorrect.

### A screen shot of a computer program AI-generated content may be incorrect.

### A screen shot of a computer code AI-generated content may be incorrect.

### 

### A screenshot of a computer program AI-generated content may be incorrect.

### Stack & Heap in C#

### [**Stack**](https://www.shiksha.com/online-courses/articles/all-that-you-need-to-know-about-stack/) is a data structure that operates in a Last In, First Out (LIFO) manner. This means that the last element added to the stack will be the first one to be removed.

### Scope is local and gets allocated when method is invoked.It gets cleared when method cleared.

### Value types are stored in stack. Eg: int, float ,double etc..

### Heap memory:

### All dynamic requirements are fulfilled here.After usage of the variable, it is returned to heap for further use.

### Reference types are stored in heap memory. Eg: string, object

### A screenshot of a computer screen AI-generated content may be incorrect.

### Finally vs Finalize vs Dispose

### 

### A screenshot of a computer program AI-generated content may be incorrect.

### A close-up of a document AI-generated content may be incorrect.

### A screenshot of a chat AI-generated content may be incorrect.