

1.What is an Abstract Class?

An abstract class is an incomplete class or special class we can't be instantiated. The purpose of an abstract class is to provide a blueprint for derived classes and set some rules what the derived classes must implement when they inherit an abstract class.

2. What is Interface?

Interface in C# is a blueprint of a class. It is like abstract class because all the methods which are declared inside the interface are abstract methods. It cannot have method body and cannot be instantiated. It is used to achieve multiple inheritance which can't be achieved by class.

3. What are access Specifiers?

Access specifiers define how the members (attributes and methods) of a class can be accessed.

There are three types of access specifiers. They are:

- public - members are accessible from outside the class
- private - members cannot be accessed (or viewed) from outside the class
- protected - members cannot be accessed from outside the class, however, they can be accessed in inherited classes. You will learn more about Inheritance later.

4. What are get and set properties?

The get method returns the value of the variable name. The set method assigns a value to the name variable. The value keyword represents the value we assign to the property.

5. What is Inheritance?

Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. Hence, inheritance facilitates Reusability and is an important concept of OOPs.

6. What is Overloading?

Overloading happens when you have two methods with the same name but different signatures (or arguments). In a class we can implement two or more methods with the same name. Overloaded methods are differentiated based on the number and type of parameter passed as arguments to the methods.

7. What is Overriding?

If derived class defines same method as defined in its base class, it is known as method overriding in C#. It is used to achieve runtime polymorphism. It enables you to provide specific implementation of the method which is already provided by its base class.

To perform method overriding in C#, you need to use virtual keyword with base class method and override keyword with derived class method.

8. What are Extension Methods?

Extension methods enable you to "add" methods to existing types without creating a new derived type, recompiling, or otherwise modifying the original type. Extension methods are static methods, but they're called as if they were instance methods on the extended type.

9. What are New Features of C# 8.0?

- Record structs
- Improvements of structure types
- Interpolated string handlers
- global using directives
- File-scoped namespace declaration
- Extended property patterns
- Improvements on lambda expressions
- Allow const interpolated strings
- Record types can seal ToString()
- Improved definite assignment
- Allow both assignment and declaration in the same deconstruction
- Allow AsyncMethodBuilder attribute on methods
- CallerArgumentExpression attribute
- Enhanced #line pragma

10. what is an constructor in c#?

A constructor is a method whose name is the same as the name of its type. Its method signature includes only an optional access modifier, the method name and its parameter list; it does not include a return type.

11. What is Collection Framework in c#?

C# collection types are designed to store, manage and manipulate similar data more efficiently. Data manipulation includes adding, removing, finding, and inserting data in the collection. Collection types implement the following common functionality: Adding and inserting items to a collection.

12. What are generics?

Generic is a concept that allows us to define classes and methods with placeholder. C# compiler replaces these placeholders with specified type at compile time. The concept of generics is used to create general purpose classes and methods.

To define generic class, we must use angle <> brackets. The angle brackets are used to declare a class or method as generic type. In the following example, we are creating generic class that can be used to deal with any type of data.

13. what is arrayList in c#?

In C#, the ArrayList is a non-generic collection of objects whose size increases dynamically. It is the same as Array except that its size increases dynamically. An ArrayList can be used to add unknown data where you don't know the types and the size of the data.

14. What is List<T>?

The List<T> is a collection of strongly typed objects that can be accessed by index and having methods for sorting, searching, and modifying list. It is the generic version of the ArrayList that comes under System.

15. What is difference between String & StringBuilder?

String: String is immutable, Immutable means if you create string object then you cannot modify it and It always create new object of string type in memory.

Example :

```
string strMyValue = "Hello Visitor";  
  
// create a new string instance instead of changing the old one  
  
strMyValue += "How Are";  
  
strMyValue += "You ??";
```

StringBuilder: StringBuilder is mutable, means if create string builder object then you can perform any operation like insert, replace or append without creating new instance for every time.it will update string at one place in memory doesnt create new space in memory.

Example:

```
StringBuilder sbMyValue = new StringBuilder("");  
  
sbMyValue.Append("Hello Visitor");  
  
sbMyValue.Append("How Are You ??");  
  
string strMyValue = sbMyValue.ToString();
```

16. What is difference between public and internal?

public is visible from wherever. internal is visible only within an assembly.

internal is useful when you want to declare a member or type inside a DLL, not outside that. Normally, when you declare a member as public , you can access that from other DLLs. But, if you need to declare something to be public just inside your class library, you can declare it as internal .

1.What is MVC?

MVC stands for Model, View, and Controller. MVC separates an application into three components - Model, View, and Controller.

Model: Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.

View: View in MVC is a user interface. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

Controller: The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

2. What is Design Pattern?

Design patterns are solutions to software design problems you find again and again in real-world application development. Patterns are about reusable designs and interactions of objects.

Design Patterns are categorized mainly into three categories: Creational Design Pattern, Structural Design Pattern, and Behavioral Design Pattern. These are differed from each other on the basis of their level of detail, complexity, and scale of applicability to the entire system being design.

3. What is Http Status Codes?

An HTTP status code is a server response to a browser's request. When you visit a website, your browser sends a request to the site's server, and the server then responds to the browser's request with a three-digit code: the HTTP status code.

- 1xx informational response – the request was received, continuing process
- 2xx successful – the request was successfully received, understood, and accepted
- 3xx redirection – further action needs to be taken in order to complete the request
- 4xx client error – the request contains bad syntax or cannot be fulfilled
- 5xx server error – the server failed to fulfil an apparently valid request

4. what is request pipeline in .net core?

The Request Pipeline is the mechanism by which requests are processed beginning with a Request and ending with a Response.

The pipeline specifies how the application should respond to the HTTP request. The Request arriving from the browser goes through the pipeline and back

The individual components that make up the pipeline are called Middleware.

5. What is Configure services and configure method?

The Configure method is used to specify how the app responds to HTTP requests. The request pipeline is configured by adding middleware components to an `IApplicationBuilder` instance. `IApplicationBuilder` is available to the Configure method, but it isn't registered in the service container.

The `ConfigureServices` method is a place where you can register your dependent classes with the built-in IoC container. After registering dependent class, it can be used anywhere in the application. You just need to include it in the parameter of the constructor of a class where you want to use it.

6.What is Web CLI? Why is it used in Dotnet Core?

The .NET command-line interface (CLI) is a cross-platform toolchain for developing, building, running, and publishing .NET applications.

CLI commands:

Basic commands

- new
- restore
- build
- publish
- run
- test
- vstest
- pack
- migrate
- clean
- sln
- help

- store

Project modification commands

- add package
- add reference
- remove package
- remove reference
- list reference

Advanced commands

- nuget delete
- nuget locals
- nuget push
- msbuild
- dotnet install script

Tool management commands

- tool install
- tool list
- tool update
- tool restore Available since .NET Core SDK 3.0.
- tool run Available since .NET Core SDK 3.0.
- tool uninstall

7. What is Dotnet Core?

.NET Core is a new version of .NET Framework, .NET Core is a general-purpose software development framework, it easily allows developers to build a plethora of software including Web, Desktop, Mobile, Cloud, Gaming, IoT, etc. The applications built with .NET Core are supported on Windows, Linux, and macOS.

8.WHAT IS EF DOTNET CORE EF?

Entity Framework (EF) Core is a lightweight, extensible, open source and cross-platform version of the popular Entity Framework data access technology. EF Core can serve as an object-relational mapper (O/RM), which: Enables .NET developers to work with a database using .

9.WHAT IS SWAGGER AND WHY IT IS USED?

Swagger allows you to describe the structure of your APIs so that machines can read them.

Swagger provides a set of great tools for designing APIs and improving the work with web services:

10.WHAT IS MIDDLEWARE IN DOTNET CORE?

Middleware are software components that are assembled into an application pipeline to handle requests and responses. Each component chooses whether to pass the request on to the next component in the pipeline, and can perform certain actions before and after the next component is invoked in the pipeline.

11.WHAT IS DI? WHAT IS IOC?

Inversion of Control(IoC) is also known as Dependency injection (DI).

IoC - Inversion of control is generic term, independent of language, it is actually not create the objects but describe in which fashion object is being created.

DI - Dependency Injection is concrete term, in which we provide dependencies of the object at run time by using different injection techniques

12.What is HOSTING MODEL?

The three types of web hosting models to consider are shared hosting, virtual private servers, and dedicated physical servers

13.CONFIGURESERVICES METHOD IN STARTUP CS?

The ConfigureServices method is a place where you can register your dependent classes with the built-in IoC container. After registering dependent class, it can be used anywhere in the application. You just need to include it in the parameter of the constructor of a class where you want to use it.

14.What is ROUTING?

Routing is the process of selecting a path for traffic in a network or between or across multiple networks

15.What is KESTREL SERVER?

Kestrel is an open source, cross platform, light weight and a default webserver used for Asp.Net Core applications. Asp.Net Core applications run Kestrel webserver as in-process server to handle web request. Kestrel webserver is based on async I/O library called libuv primarily developed for Node. js.

16.What are TAG HELPERS?

Tag helpers are reusable components for automating the generation of HTML in Razor Pages. Tag helpers target specific HTML tags. The ASP.NET Core framework includes a number of predefined tag helpers targeting many commonly used HTML elements as well as some custom tags: Anchor tag helper. Cache tag helper.

17.What is STATE MANAGEMENT?

It refers to managing the state of one or several user interface control systems like that of radio controls, action buttons, input fields, etc. It is a programming technique for User Interface in which the state of a single UI control completely or partially depends on the state of all the other UI controls.

18.What is BUILT-IN MIDDLEWARE?

Middleware are software components that are assembled into an application pipeline to handle requests and responses. Each component chooses whether to pass the request on to the next component in the pipeline, and can perform certain actions before and after the next component is invoked in the pipeline.

19.What is CONTENT NEGOTIATION?

In HTTP, content negotiation is the mechanism that is used for serving different representations of a resource to the same URI to help the user agent specify which representation is best suited for the user (for example, which document language, which image format, or which content encoding).

20.What is CROSS PLATFORM?

In computing, cross-platform software (also called multi-platform software, platform-agnostic software, or platform-independent software) is computer software that is designed to work in several computing platforms.

21.What is CACHING?

Caching is a technique that stores a copy of a given resource and serves it back when requested. When a web cache has a requested resource in its store, it intercepts the request and returns a copy of the stored resource instead of redownloading the resource from the originating server.

22.What are COOKIES?

A cookie is a piece of data typically used to store information about the user and is stored on the user's computer.

23.What is an ENDPOINT?

a point marking the completion of a process or stage of a process especially.

24.What are REST ENDPOINTS?

A REST Service Endpoint is an endpoint which services a set of REST resources. The URI for REST Service Endpoints entities.

25.What are HTTP HEADERS?

An HTTP header is a field of an HTTP request or response that passes additional context and metadata about the request or response. For example, a request message can use headers to indicate it's preferred media formats, while a response can use header to indicate the media format of the returned body

26. What are Return type in API Actions?

A Web API controller action can return any of the following: void. HttpResponseMessage. IHttpActionResult.

There are 7 types of content returning results:

- ViewResult.

- PartialViewResult.
- ContentResult.
- EmptyResult.
- FileResult.
- JsonResult.
- JavaScriptResult.

27.What are CORS?

Cross-Origin Resource Sharing

Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources

28.What is ORM?

Object-relational mapping (ORM) is a programming technique in which a metadata descriptor is used to connect object code to a relational database.

29. What is App Life Cycle of Dotnet Core Application?

The ASP.NET Core MVC Request Life Cycle is a sequence of events, stages or components that interact with each other to process an HTTP request and generate a response that goes back to the client.

When an ASP.NET page executes, it undergoes a life cycle that includes various stages. The important stages in the page life cycle are Initialization, Load, Control Events, Rendering and Unload.

30. What are different Configuration Sources we have in Dotnet core?

Configuration in ASP.NET Core is performed using one or more configuration providers. Configuration providers read configuration data from key-value pairs using a variety of configuration sources: Settings files, such as appsettings.

31. How do We manage Session in Dotnet Core Application?

Session state is an ASP.NET Core scenario for storage of user data while the user browses a web app. Session state uses a store maintained by the app to persist data across requests from a client.

1. Launch the Visual Studio 2017 IDE.
2. Click on File > New > Project.
3. Select “ASP.Net Core Web Application (.Net Core)” from the list of templates displayed.
4. Specify a name for the project.
5. Click OK to save the project.
6. Select “API” in the “New .Net Core Web Application...” window.
7. Select “.Net Core” as the runtime and ASP.Net Core 2.1 (or later) from the drop-down list of controls at the top.
8. Select “Web Application (Model-View-Controller)” as the project template.
9. Uncheck the “Enable Docker Support” box.
10. Ensure that “No Authentication” is selected as we won’t be using authentication here.
11. Ensure that “Configure for HTTPS” is unchecked as we won’t need HTTPS either. (See the figure below.)
12. Click OK.