Senior .Net Developer Job Interview Questions And Answers

Interview Questions Answers

https://interviewquestionsanswers.org/

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Senior .Net Developer will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit Senior .Net Developer Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Senior .Net Developer category. To ensure quality, each submission is checked by our team, before it becomes live. This Senior .Net Developer Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

Senior .Net Developer Interview Questions And Answers Guide.

Question - 1:
Tell us what is the Native Image Generator?

Ans:
It is a tool that compiles the .Net assemblies to machine code for a specific processor. In this way, it improves its performance since the JIT no longer intervenes.
View All Answers

Question - 2:
Tell us what is a sealed class?

Ans:
It is a class that is not inheritable. A sealed class comes in use for a super specialized class, by design, and prevents modification by overwriting.
View All Answers

Question - 3:
Please explain what is Reflection and what is it for?

Ans:
It is the ability to read, instantiate, and invoke the properties & methods of an assembly's classes. It is especially useful when we do not have the source code for classes, only their assembly.
View All Answers

Question - 4:
Please explain is the JIT an interpreter?

Ans:
No, the JIT is not an interpreter. It is a compiler at runtime that improves performance compiling method by method only once. If the method is called a new account, the native code already compiled is used. However, an interpreter executes the same every block of code.
View All Answers

Question - 5:
Tell us what is a variable of implicit type and what is its scope?

Ans:
It is a variable that doesn't require type declaration since the compiler automatically determines its type. Its scope is local, within a method. It only allows inferring the kind the first time it assigns a value to the second. However, if the type is different, it throws an error.
View All Answers

Question - 6:
Please explain what is a delegate?

Ans:
It is the definition of a method that encapsulates certain arguments and type of return. It allows passing a method as an argument of a function, as long as it matches its specific signature.
View All Answers

Question - 7:
Please explain what is the difference between a class and an object?

Ans:

In short, a class is the definition of an object, and an object is instance of a class.
We can look at the class as a template of the object: it describes all the properties, methods, states and behaviors that the implementing object will have. As mentioned, an object is an instance of a class, and a class does not become an object until it is instantiated. There can be more instances of objects based on the one class, each with different properties.

View All Answers

Question - 8:

Explain me difference between public and static modifiers?

Ans:

To invoke a method, field, or static property, you don't need to instantiate the class.
On the other hand, to invoke a public method, you need an instance of a class.

View All Answers

Question - 9:

What is the concept of inheritance and how it works in .NET?

Ans:

In general OOP terms, inheritance means that a class can be based on another class, with the child class taking on the attributes of the parent class. For example, coders can create a class called Vehicle, and then child classes called Truck, Car and Motorcycle - all of which inherit the attributes of Vehicle.
To demonstrate their understanding of the interview question and the framework, candidates may bring up how .NET supports single inheritance only, which means that a class can inherit only from one other class. Their answer may also touch on the transitive nature of inheritance - for example, the Ford class inherits from Car, which inherits from Vehicle.

View All Answers

Question - 10:

Please explain what are an object and a class?

Ans:

An object is an instance of a class, and a class is a template for creating objects. Class is the definition of an object, the description of its characteristics and operations, its properties and its methods. An object has an identity because its characteristics have values.

View All Answers

Question - 11:

Explain me what is a design pattern and what is it for?

Ans:

It is a reusable template to solve a common problem at the design level. It is not the code but best practices to codify a solution. Some examples are Singleton, Abstract Factory, Observer or Pub/Sub, Model View Controller, Model View Presenter, and Model-View View-Model.

View All Answers

Question - 12:

Explain me what is the difference between an abstract class and an interface?

Ans:

An abstract class is always used as a base class. It provides some abstract/virtual members that the inheriting entities must implement, as well as a partial implementation for a functionality. For extra credit, job candidates might mention that this class can also declare fields. Developers cannot create an object from this class.
An interface, on the other hand, can declare properties, methods and events only (no access modifiers). The developer must implement all declared members. In short, an interface designates a contract/behavior that implementing classes should have.

View All Answers

Question - 13:

What is the Single Responsibility Principle?

Ans:

Single responsibility is the concept of a Class doing one specific thing (responsibility) and not trying to do more than it should, which is also referred to as High Cohesion.
Classes don't often start out with Low Cohesion, but typically after several releases and different developers adding onto them, suddenly you'll notice that it became a monster or God class as some call it. So the class should be refactored.

View All Answers

Question - 14:

Please explain what are an inheritance, polymorphism, and encapsulation?

Ans:

Inheritance is the ability to reuse definitions from one class in another and to base one class on another.
Polymorphism helps declare the same method within a class with different argument or return types.
Encapsulation is to be able to expose only the methods, property and arguments necessary to use the operations of a class. However, the detailed implementation remains private, hidden to other objects.

View All Answers

Question - 15:

Please explain what is immutability, what is it for and how is it codified?

Ans:

The ability of objects not to change their state, once created, helps improve the maintainability of the code. When a mutable object encapsulates its changes without being explicit in the code, following the flow becomes difficult. The level of difficulty increases in case of multi-threaded applications. To create immutable objects, pass the arguments for their creation in the constructor; make their properties read-only later.

View All Answers

Question - 16:

Tell us the differences between an Interface and an Abstract Class in .NET?

Ans:

An interface merely declares a contract or a behavior that implementing classes should have. It may declare only properties, methods, and events with no access modifiers. All the declared members must be implemented.
An abstract class provides a partial implementation for a functionality and some abstract/virtual members that must be implemented by the inheriting entities. It can declare fields too.
Neither interfaces nor abstract classes can be instantiated.

View All Answers

Question - 17:

Please explain when should you use .NET Web Forms over ASP.NET MVC?

Ans:

Traditionally, the .NET Framework has been based on Web Forms. This was essentially an effort to create web services using Microsoft's existing Visual Studio Tools without forcing developers to learn new scripting languages. Web Forms still allows developers to create quick and simple applications, and some legacy systems may still run as Web Forms.
ASP.NET MVC is increasingly the standard for contemporary developers, however. In a .NET interview, a strong candidate should be able to highlight the advantages of the Model-View-Controller (MVC) architectural pattern. MVC's most important feature is that it allows applications to be broken down into discrete models, views and controllers, making them much easier to test during development.

View All Answers

Question - 18:

Do you know the difference between the Stack and the Heap?

Ans:

The short answer would be: in the Stack are stored value types (types inherited from System.ValueType), and in the Heap are stored reference types (types inherited from System.Object).
We can say the Stack is responsible for keeping track of what is actually executing and where each executing thread is (each thread has its own Stack). The Heap, on the other hand, is responsible for keeping track of the data, or more precise objects.

View All Answers

Question - 19:

Explain me how does LINQ work?

Ans:

Internally build the correct query (in the case of databases) or generate the corresponding operations on the collections or parse the XML and returns the relevant data. It encapsulates all these behaviors and provides a single implementation, in this way, we can use the same queries, the same language, independently of the underlying data source.

View All Answers

Question - 20:

Explain what is Mutex?

Ans:

Threads share a mutually exclusive resource manager, Mutex. It ensures that only one thread at a time makes use of one resource (one object) at a time. It is like a moderator that controls the microphone and gives the word to one person at a time. Thus, Mutex grants access to resources one thread at a time. For this, it puts the threads that want to access resources "on hold" until those are in use.

View All Answers

Question - 21:

Tell us what is Heap and what is Stack?

Ans:

* Both are memory locations, wherein Heap is global and Stack is local.
* The Heap is application level while the Stack is thread-level.
* The Stack has a defined first-in-first-out stack structure, while the Heap does not have a defined data structure.
* Its size is defined at the time of its creation. For Heap, the size is defined when starting the application and for Stack, when creating a thread.
* Both can grow dynamically.
* The Stack is faster than the Heap. A stack is in "cache" and doesn't have to synchronize with other threads like the Heap.
* The Stack stores values while the Heap stores objects.

View All Answers

Question - 22:

Explain me what is encapsulation?

Ans:

Encapsulation is one of four basic features of OOP and refers to the inclusion within a program object of methods and data needed for the object to function. For .NET interview questions like this, candidates should mention that encapsulation helps keep data from unwanted access through binding code and data in an object, which is the basic, single self-contained unit of a system.
Another way of understanding encapsulation is to think of it as "hiding" the state of an object as private or protected. Under this principle of information hiding, the internal workings of an object are segregated from the rest of the application. This is useful because it makes it less likely that other objects can modify the state or behavior of the object in question.

View All Answers

Question - 23:

Please explain what do the terms "boxing" and "unboxing" mean?

Ans:

This question can reveal how much candidates know about data types and OOP principles. The idea is relatively simple: Boxing is a process that converts a value type to an object type - by "boxing" the variable inside a dedicated object or interface. Unboxing extracts this value and stores it in a value type. Boxing was essential in some old Collection types such as ArrayList, and can still be used for accurate conversion of types - for example, from a double to an int.

View All Answers

Question - 24:

Explain me what is .NET web service?

Ans:

Web services are reusable components that allow developers to publish an application's function over the internet to make it accessible and directly interactable with other applications and objects online. Web services communicate by using standard web protocols and data formats - including HTTP, XML and SOAP - allowing them to connect across different platforms and programming languages. ASP.NET provides a simple way to develop web services. The .NET Framework provides built-in classes for building and consuming web services.

View All Answers

Question - 25:

Please explain the difference between constants and read-only variables?

Ans:

While constants and read-only variable share many similarities, there are some important differences:
* Constants are evaluated at compile time, while the read-only variables are evaluated at run time.
* Constants support only value-type variables (the only exception being strings), while read-only variables can hold reference-type variables.
* Constants should be used when the value is not changing during run time, and read-only variables are used mostly when their actual value is unknown before run time.
* Read-only variables can only be initialised at the time of declaration or in a constructor.

View All Answers

Question - 26:

Do you know what is lambda expressions in C#?

Ans:

A lambda expression is an anonymous function that you can use to create delegates or expression tree types. By using lambda expressions, you can write local functions that can be passed as arguments or returned as the value of function calls. Lambda expressions are particularly helpful for writing LINQ query expressions.

View All Answers

Question - 27:

Please explain what is the difference between constants and read-only variables?

Ans:

For constants, the compilation contains declaration and initialization. Its value cannot change. For read-only variables, the runtime execution contains the assignment of values.

View All Answers

Question - 28:

Do you know what is LINQ?

Ans:

It is standardization to consult data and convert it into objects, regardless of the source. It is a query manager for databases, XML and enumerable collections using a single language.

View All Answers

Question - 29:

What is the difference between managed and unmanaged code?

Ans:

.NET interview questions like this allow candidates to demonstrate their understanding of Common Language Runtime (CLR), a crucial part of the .NET Framework. Code written in C# or Visual Basic .NET will, when compiled, run only in the CLR, which provides functionalities such as garbage collection and memory

management. The advantage of this is that managed code is platform-independent because it runs in the CLR rather than the operating system of the machine accessing the application.
Code written in other languages, such as C or C++, produces unmanaged code, meaning developers can't rely on the CLR to provide this kind of portability. Managed and unmanaged code are interoperable. Examples of unmanaged code used in .NET include COM components, ActiveX interfaces and Win32 API functions.

View All Answers

Question - 30:

Please explain what garbage collection is and how it works. Provide a code example of how you can enforce garbage collection in .NET?

Ans:

Garbage collection is a low-priority process that serves as an automatic memory manager which manages the allocation and release of memory for the applications. Each time a new object is created, the common language runtime allocates memory for that object from the managed Heap. As long as free memory space is available in the managed Heap, the runtime continues to allocate space for new objects. However, memory is not infinite, and once an application fills the Heap memory space, garbage collection comes into play to free some memory. When the garbage collector performs a collection, it checks for objects in the managed Heap that are no longer being used by the application and performs the necessary operations to reclaim the memory. Garbage collection will stop all running threads, it will find all objects in the Heap that are not being accessed by the main program and delete them. It will then reorganize all the objects left in the Heap to make space and adjust all the Pointers to these objects in both the Stack and the Heap.
To enforce garbage collection in your code manually, you can run the following command (written in C#):
System.GC.Collect();

View All Answers

Question - 31:

Please explain what is the .Net framework and how does it work?

Ans:

It is a virtual machine that executes a managed code. The code is compiled from C# or VB .NET and is executed by the CLR (Common Language Runtime).
Its working is as follows:
* You create a program in C # or VB.Net and compile it. The code is then translated to CIL (Common Intermediate Language).
* The program is assembled into bytecode to generate a CLI (Common Language Infrastructure) assembly file of.exe or .dll format.
* When you run the program (or the DLL), it is executed by the .Net framework CLR (Common Language Runtime). Since the code isn't directly run by the operating system, it is called "Managed Code".
* The .Net Framework CLR, through the JIT (Just-In-Time) Compiler, is responsible for compiling this code managed in the intermediate language. The compiled code is then sent to the native machine language assembler for the CPU to execute it.

View All Answers

Question - 32:

Please explain what is the difference between ODBC and ADO?

Ans:

Open Database Connectivity is a standard for managing database operations in applications. The standard uses the same methods for Oracle as for Mysql. For example, it declares the connection with particularity at the user or operating system level.
ADO is a set of .Net libraries for data management, including ODBC connections. For ADO, ODBC is a driver.

View All Answers

Question - 33:

Please explain what is Heap and what is Stack?

Ans:

* Both are memory locations, wherein Heap is global and Stack is local.
* The Heap is application level while the Stack is thread-level.
* The Stack has a defined first-in-first-out stack structure, while the Heap does not have a defined data structure.
* Its size is defined at the time of its creation. For Heap, the size is defined when starting the application and for Stack, when creating a thread.
* Both can grow dynamically.
* The Stack is faster than the Heap. A stack is in "cache" and doesn't have to synchronize with other threads like the Heap.
* The Stack stores values while the Heap stores objects.

View All Answers

Question - 34:

Do you know what's the difference between .NET and Laravel?

Ans:

This is one of the more straightforward .NET interview questions you can ask. Most web developers will at least be familiar with alternative frameworks such as Laravel and should be able to discuss some of the differences between those platforms and .NET. This conversation will allow you to dig down and discover where candidates' interests lie and how they view the role of .NET developer. For example, they may focus on:
* Languages (e.g. C# support in .NET versus Laravel's support for PHP)
* Security
* Processing overheads
* .NET's integration with Visual Studio
* Third-party libraries
* Open-source community support

View All Answers

Question - 35:

Tell us what is JSON data, and what is one way that .NET developers can work with JSON?

Ans:

JSON (JavaScript Object Notation) provides developers with a way to organize and store data so it's easy to access and read. JSON is important for developers because it allows them to manipulate JSON feeds from other sites and to load them more quickly and easily than via SML/RSS feeds. Json.NET provides a way for .NET developers to define classes that parse objects and arrays from JSON text. You can also use Json.NET if you need to serialize value types into JSON text. Json.NET runs on .NET2, .NET3 and .NET4.

View All Answers

Question - 36:

What is deferred execution vs. immediate execution in LINQ?

Ans:

In LINQ, deferred execution simply means that the query is not executed at the time it is specified. Specifically, this is accomplished by assigning the query to a variable. When this is done, the query definition is stored in the variable but the query is not executed until the query variable is iterated over. For example:

```
DataContext productContext = new DataContext();
var productQuery = from product in productContext.Products
      where product.Type == "SOAPS"
      select product;   // Query is NOT executed here
foreach (var product in productQuery)   // Query executes HERE
{
  Console.WriteLine(product.Name);
}
```

You can also force immediate execution of a query. This can be useful, for example, if the database is being updated frequently, and it is important in the logic of your program to ensure that the results you're accessing are those returned at the point in your code where the query was specified. Immediate execution is often forced using a method such as Average, Sum, Count, List, ToList, or ToArray. For example:

```
DataContext productContext = new DataContext();
var productCountQuery = (from product in productContext.Products
      where product.Type == "SOAPS"
      select product).Count();   // Query executes HERE
```

View All Answers

Question - 37:

What is the difference between boxing and unboxing?

Ans:

Boxing is the process of converting a value type to the type object, and unboxing is extracting the value type from the object. While the boxing is implicit, unboxing is explicit.
Example (written in C#):

```
int i = 13;
object myObject = i;  // boxing
i = (int)myObject; // unboxing
```

View All Answers

Question - 38:

What is the difference between Task and Thread in .NET?

Ans:

* Thread represents an actual OS-level thread, with its own stack and kernel resources. Thread allows the highest degree of control; you can Abort() or Suspend() or Resume() a thread, you can observe its state, and you can set thread-level properties like the stack size, apartment state, or culture. ThreadPool is a wrapper around a pool of threads maintained by the CLR.
* The Task class from the Task Parallel Library offers the best of both worlds. Like the ThreadPool, a task does not create its own OS thread. Instead, tasks are executed by a TaskScheduler; the default scheduler simply runs on the ThreadPool. Unlike the ThreadPool, Task also allows you to find out when it finishes, and (via the generic Task) to return a result.

View All Answers

Question - 39:

Please explain what is the difference between encrypting a password and applying a hashing?

Ans:

It is quite difficult (almost impossible) to decrypt a hashing (MD5 or SHA1, for example). The process of password validation compares the password in plain text with a hash to the stored one.
Conversely, one can decrypt an encrypted password with access to the keys and the encryption algorithms (such as Triple-DES).

View All Answers

Question - 40:

Tell me what is serialization?

Ans:

Serialization converts an object to a data stream. However, for this, you must implement ISerialize.

View All Answers

Question - 41:

Explain me what is the .Net framework and how does it work?

Ans:

It is a virtual machine that executes a managed code. The code is compiled from C# or VB .NET and is executed by the CLR (Common Language Runtime).
Its working is as follows:
* You create a program in C # or VB.Net and compile it. The code is then translated to CIL (Common Intermediate Language).
* The program is assembled into bytecode to generate a CLI (Common Language Infrastructure) assembly file of.exe or .dll format.
* When you run the program (or the DLL), it is executed by the .Net framework CLR (Common Language Runtime). Since the code isn't directly run by the operating system, it is called "Managed Code".
* The .Net Framework CLR, through the JIT (Just-In-Time) Compiler, is responsible for compiling this code managed in the intermediate language. The compiled code is then sent to the native machine language assembler for the CPU to execute it.

View All Answers

Question - 42:

As you know read-only variables and constants have many similarities, but what is at least one way that they differ?

Ans:

Here are two possible answers to .NET interview questions of this nature:
* Read-only variables can support reference-type variables. Constants can hold only value-type variables.
* Developers evaluate read-only variables at the runtime. They evaluate constants at the compile time.

View All Answers

Question - 43:

Explain me what is the difference between a class and an object, and how do these terms relate to each other?

Ans:

A class is a comprehensive data type that is the primary building block, or template, of OOP. Class defines attributes and methods of objects, and contains an object's behavior and data. An object, however, represents an instance of class. As a basic unit of a system, objects have identity and behavior as well as attributes.
Make sure candidates respond to the second part of this .NET interview question, addressing how these terms are related to each other. Answer: The relationship is based on the fact that a class defines the states and properties that are common to a range of objects.

View All Answers

Question - 44:

Tell us what do the following acronyms in .NET stand for: IL, CIL, MSIL, CLI and JIT?

Ans:

IL, or Intermediate Language, is a CPU independent partially compiled code. IL code will be compiled to native machine code using current environmental properties by Just-In-Time compiler (JIT). JIT compiler translates the IL code to an assembly code and uses the CPU architecture of the target machine to execute a .NET application. In .NET, IL is called Common Intermediate Language (CIL), and in the early .NET days it was called Microsoft Intermediate Language (MSIL).
CLI, or Common Language Infrastructure, is an open specification developed by Microsoft. It is a compiled code library used for deployment, versioning, and security. In .NET there are two CLI types: process assemblies (EXE) and library assemblies (DLL). CLI assemblies contain code in CIL, and as mentioned, during compilation of CLI programming languages, the source code is translated into CIL code rather than into platform or processor specific object code.
To summarize:
* When compiled, source code is first translated to IL (in .NET, that is CIL, and previously called MSIL).
* CIL is then assembled into a bytecode and a CLI assembly is created.
* Before code execution, CLI code is passed through the runtime's JIT compiler to generate native machine code.
* The computer's processor executes the native machine code.

View All Answers

Question - 45:

Tell us the difference between the while and for loop. Provide a .NET syntax for both loops?

Ans:

Both loops are used when a unit of code needs to execute repeatedly. The difference is that the for loop is used when you know how many times you need to iterate through the code. On the other hand, the while loop is used when you need to repeat something until a given statement is true.
The syntax of the while loop in C# is:

```
while (condition [is true])
{
  // statements
}
```

The syntax of the while loop in VB.NET is:

```
While condition [is True]
  ' statements
End While
```

The syntax of the for loop in C# is:

```
for (initializer; condition; iterator)
{
  // statements
}
```

The syntax of the for loop in VB.NET is:

```
For counter [ As datatype ] = start To end [ Step step ]
  ' statements
Next [ counter ]
```

View All Answers

Question - 46:

Explain me why do we use MSMQ?

Ans:

Microsoft Message Queuing, or MSMQ, is technology for asynchronous messaging. Whenever there's need for two or more applications (processes) to send messages to each other without having to immediately know results, MSMQ can be used. MSMQ can communicate between remote machines, even over Internet. It's free and comes with Windows, but is not installed by default.
This mainly addresses the common use case of asynchronous message processing: you have a service Service1 that communicates (send messages) with another part of your software architecture, say Service2.
Main problem: what if Service2 becomes suddenly unavailable? Will messages be lost? If you use MSMQ it won't: Service1 will send messages into a queue, and Service2 will dequeue when it is available.

View All Answers

Question - 47:

Explain me what is an anonymous method and how is it different from a lambda expression?

Ans:

For an anonymous method, the declaration comes before its use and implementation through a delegate. Also, this method doesn't require a name.
A lambda expression refers to an anonymous method in a single line, elegantly replacing the representative for this function.

View All Answers

Question - 48:

Tell us what is the difference between struct and class?

Ans:

A class is a definition of an object and is inheritable. A structure, on the other hand, defines a type of data and is non-inheritable.

View All Answers

Question - 49:

Tell me what is a "jagged array"?

Ans:

It is an arrangement of arrangements.

View All Answers

Question - 50:

Explain me what are the deferred execution and the immediate execution in LINQ?

Ans:

A deferred execution encapsulates a query's definition without executing it till the data is used at runtime. However, an immediate implementation calls the query at the same moment of its definition.
By default, the executions are deferred but we can do them immediately by calling "To List ()". For example, in this way, a list of objects will be executed and returned to us when we define it.

View All Answers

Question - 51:

Do you know what are three common acronyms used in .NET, and what do they stand for?

Ans:

This one should be easy for .NET developer candidates to answer. The question allows them some flexibility in choosing terms with which they are most familiar. Three frequently used acronyms in .NET are IL, CIL and CLI:
* IL stands for Intermediate Language, which is an object-oriented programming language that is a partially compiled code that .NET developers will then compile to native machine code.
* CIL stands for Common Intermediate Language, formerly known as Microsoft Intermediate Language (MSIL). This is another programming language that .NET developers use, and it represents the lowest possible level for a language that humans can still read.
* CLI stands for Common Language Infrastructure. This is a compiled code library that Microsoft developed as an open specification. Developers use CLI for security, versioning and deployment purposes.

View All Answers

Question - 52:

What is the difference between a stack and a queue?

Ans:

This .NET interview question tests candidates' basic knowledge of collections. Along with stacks and queues in this category are hash tables, bags, dictionaries and lists. A stack keeps track of what is executing and contains stored value types to be accessed and processed as LIFO (Last-In, First-Out), with elements inserted and deleted from the top end.
A queue, on the other hand, lists items on a FIFO (First-In, First-Out) basis in terms of both insertion and deletion, with items inserted from the rear end and deleted from the front end of the queue.

View All Answers

Question - 53:

What is implement a generic action in WebAPI?

Ans:

It's not possible, as the WebAPI runtime needs to know the method signatures in advance.

View All Answers

Question - 54:

Please explain what inheritance is, and why it's important?

Ans:

Inheritance is one of the most important concepts in object-oriented programming, together with encapsulation and polymorphism. Inheritance allows developers to create new classes that reuse, extend, and modify the behavior defined in other classes. This enables code reuse and speeds up development. With inheritance, developers can write and debug one class only once, and then reuse that same code as the basis for the new classes. The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class. By default, all classes in .NET are inheritable.

View All Answers

Question - 55:

Tell us the difference between managed and unmanaged code?

Ans:

Managed code is a code created by the .NET compiler. It does not depend on the architecture of the target machine because it is executed by the CLR (Common Language Runtime), and not by the operating system itself. CLR and managed code offers developers few benefits, like garbage collection, type checking and exceptions handling.
On the other hand, unmanaged code is directly compiled to native machine code and depends on the architecture of the target machine. It is executed directly by the operating system. In the unmanaged code, the developer has to make sure he is dealing with memory usage and allocation (especially because of memory leaks), type safety and exceptions manually.
In .NET, Visual Basic and C# compiler creates managed code. To get unmanaged code, the application has to be written in C or C++.

View All Answers

Question - 56:

Do you know what is .NET Standard?

Ans:

* .NET Standard solves the code sharing problem for .NET developers across all platforms by bringing all the APIs that you expect and love across the environments that you need: desktop applications, mobile apps & games, and cloud services
* .NET Standard is a set of APIs that all .NET platforms have to implement. This unifies the .NET platforms and prevents future fragmentation.
* .NET Standard 2.0 will be implemented by .NET Framework, .NET Core, and Xamarin. For .NET Core, this will add many of the existing APIs that have been requested.
* .NET Standard 2.0 includes a compatibility shim for .NET Framework binaries, significantly increasing the set of libraries that you can reference from your .NET Standard libraries.
* .NET Standard will replace Portable Class Libraries (PCLs) as the tooling story for building multi-platform .NET libraries.

View All Answers

Question - 57:

Tell us why do we use the "using" statement?

Ans:

We use the "using" statement to make sure that we release the resources of the object in use. It always calls "Dispose of" when it finishes its block of code.

View All Answers

Question - 58:

Please tell us what is the difference between Override and Overload in a method?

Ans:

Override is to overwrite the method with the same signature (parameters and return type) but different functionality. Overwriting requires a "virtual" declaration of the method.
On the other hand, overloading refers to coding several versions of the same method. Though the "virtual" declaration for a method is not necessary to overload, it requires a different signature (parameters and/or return value).

View All Answers

Question - 59:

Do you know what is the difference between an abstract class and an interface?

Ans:

* An abstract class can contain both public and private constructors, methods, and fields. On the contrary, the interface contains only methods and public properties.
* You can only inherit from an abstract class, but implement many interfaces.
* An interface defines behavior, something that the class that implements it can do. Contrary, an abstract class defines what the class is and what it represents.
* You can't instantiate anyone.
* An abstract class is useful when creating components, making a partial initial implementation and a specific definition. This leaves you free to implement other methods.

View All Answers

Question - 60:

Do you know what is Garbage Collector?

Ans:

Garbage Collector is an automatic process of memory release. When memory goes low, it goes through the Heap and eliminates the objects no longer in use. It frees up memory, reorganizes remaining threads and adjusts pointers to these objects, both in Heap and Stack.

View All Answers

Microsoft .Net Technologies Most Popular & Related Interview Guides

1 : **MSF Interview Questions and Answers.**

2 : .Net Architecture Interview Questions and Answers.

3 : ASP.Net MVC Interview Questions and Answers.

4 : Entity Framework Interview Questions and Answers.

5 : C# (Sharp) Programming Language Interview Questions and Answers.

6 : VB .Net Interview Questions and Answers.

7 : ADO.NET Interview Questions and Answers.

8 : WCF (Windows Communication Foundation) Interview Questions and Answers.

9 : Crystal Reports Interview Questions and Answers.

10 : .Net Database Interview Questions and Answers.

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG