**LEVEL-2:MEDIUM**

**C# .NET**

**80 INTERVIEW QUESTIONS**

**AND**

**ANSWERS**

**Done by Babika**

**KloudOne( 3rd Batch)**

**Q -1) Explain CAS (Code Access Security)?**

.Net provides a security model that prevents unauthorized access to resources. CAS is a part of that security model. CAS is present in the CLR. It enables the users to set permissions at a granular level for the code.

CLR then executes the code depending on the available permissions. CAS can be applied only to the managed code. Unmanaged code runs without CAS. If CAS is used on assemblies, then the assembly is treated as partially trusted. Such assemblies must undergo checks every time when it tries to access a resource.

**The different components of CAS are Code group, Permissions, and Evidence.**

* **Evidence**– To decide and assign permissions, the CAS and CLR depend on the specified evidence by the assembly. The examination of the assembly provides details about the different pieces of evidence. Some common evidence include Zone, URL, Site, Hash Value, Publisher and Application directory.
* **Code Group** – Depending on the evidence, codes are put into different groups. Each group has specific conditions attached to it. Any assembly that matches those condition is put into that group.
* **Permissions** – Each code group can perform only specific actions. They are called Permissions. When CLR loads an assembly, it matches them to one of the code groups and identifies what actions those assemblies can do. Some of the Permissions include Full Trust, Everything, Nothing, Execution, Skip Verification, and the Internet.

**Q-2) Differentiate between constants and read-only variables.**

|  |  |
| --- | --- |
| **Constants** | **Read-only Variables** |
| Evaluated at compile time | Evaluated at run-time |
| Support only value type variables | They can hold the reference type variables |
| They are used when the value is not changing at compile time | Used when the actual value is unknown before the run-time |
| Cannot be initialized at the time of declaration or in a constructor | Can be initialized at the time of declaration or in a constructor |

**Q-3) What are the different versions of the .NET framework?**

|  |  |  |
| --- | --- | --- |
| **Version** | **.NET Framework** | **Visual Studio** |
| C# 1.0 | .NET Framework 1.0/1.1 | Visual Studio .NET 2002 |
| C# 2.0 | .NET Framework 2.0 | Visual Studio 2005 |
| C# 3.0 | .NET Framework 3.0/3.5 | Visual Studio 2008 |
| C# 4.0 | .NET Framework 4.0 | Visual Studio 2010 |
| C# 5.0 | .NET Framework 4.5 | Visual Studio 2012/2013 |
| C# 6.0 | .NET Framework 4.6 | Visual Studio 2013/2015 |
| C# 7.0 | .NET CORE | Visual Studio 2017 |

**Q-4) What is LINQ?**

It is an acronym for Language integrated query which was introduced with visual studio 2008. LINQ is a set of features that extend query capabilities to the .NET framework language syntax that allows data manipulation irrespective of the data source. LINQ bridges the gap between the world of objects and the world of data.

**Q-5) .Net vs Java ?**

|  |  |  |
| --- | --- | --- |
| **Feature** | **MS .Net** | **Java/JEE** |
| Operating System | Windows | Multiple OS |
| Common Language Runtime | Common Language Runtime | Java Virtual Machine |
| XML | System XML | Java API for XML Processing |
| Naming | Java Naming and Directory Interface | Active Directory Service Interfaces |
| HTTP Engine | Internet Information Services | Application Servers from various vendors |
| Server Components | .Net, COM+Services | Enterprise Java Beans |

**Q-6) What are the defining traits of an object-oriented language?**

The defining traits of an object-oriented language are:

* Inheritance
* Abstraction
* Encapsulation
* Polymorphism

**1. Inheritance**: The main class or the root class is called as a Base Class. Any class which is expected to have ALL properties of the base class along with its own is called as a Derived class. The process of deriving such a class is Derived class.

**2. Abstraction**: Abstraction is creating models or classes of some broad concept. Abstraction can be achieved through Inheritance or even Composition.

**3. Encapsulation:** Encapsulation is a collection of functions of a class and object. The “Food” class is an encapsulated form. It is achieved by specifying which class can use which members (private, public, protected) of an object.

**4. Polymorphism**: Polymorphism means existing in different forms. Inheritance is an example of Polymorphism. A base class exists in different forms as derived classes. Operator overloading is an example of Polymorphism in which an operator can be applied in different situations.

**Q-7) How do you prevent a class from being inherited?**

In C#, we can use the sealed keyword to prevent a class from being inherited.

**Q-8) What are MDI and SDI?**

* **MDI( Multiple Document Interface):** An MDI lets you open multiple windows, it will have one parent window and as many child windows. The components are shared from the parent window like menubar, toolbar, etc.
* **SDI( Single Document Interface):** It opens each document in a separate window. Each window has its own components like menubar, toolbar, etc. Therefore it is not constrained to the parent window.

**Q-9) What is the difference between custom and user control?**

|  |  |
| --- | --- |
| **Custom Control** | **User Control** |
| Derives from control | Derives from UserControl |
| Dynamic Layout | Static Layout |
| Defines a single control | Defines a set of con |
| It has full toolbox support | Cannot be added to the toolbox |
| Loosely coupled control | Tightly coupled control |

**Q-10) What is a garbage collector?**

Garbage collector feature in .NET frees the unused code objects in the memory. The memory head is divided into 3 generations:

* Generation 0: It stores short-lived objects.
* Generation 1: This is for medium-lived objects.
* Generation 2: It stores long-lived objects.

Collection of garbage refers to the collection of objects stored in the generations.

**Q-11) Explain MVC?**

MVC stands for model view controller which is an architecture to build .NET applications.

**Model:**They are the logical part of any application that handles the object storage and retrieval from the databases for an application.

**View:**View handles the UI part of an application. They get the information from the models for their display.

**Controller:**They handle the user interactions, figure out the responses for the user input and also render the view that is required for the user interaction.

**Q-12) List the events in the page life cycle?**

Following are the events in the page life cycle:

* Page\_PreInit
* Page\_Init
* Page\_InitComplete
* Page\_PreLoad
* Page\_Load
* Page\_LoadComplete
* Page\_PreRender
* Render

**Q-13) Explain localization and globalization?**

|  |  |
| --- | --- |
| **Localization** | **Globalization** |
| It means changing the already globalized application to cater to a specific language or culture. | Globalization is the process of developing applications to support multiple languages. |
| Microsoft.Extensions.Localization is used to localize the application content. | Existing applications can also be converted to support multiple languages. |

**Q-14) What is the difference between a stack and a heap?**

|  |  |
| --- | --- |
| **Stack** | **Heap** |
| Stored value type | Stored reference type |
| A stack is responsible for keeping track of each executing thread and its location. | The heap is responsible for keeping track of the more precise objects or data. |

**Q-15) What is the application domain?**

ASP.NET introduces a concept of application domain or AppDomain which is like a lightweight process that acts like both container and boundary. The .NET run-time uses the AppDomain as a container for data and code. The CLR allows multiple .NET applications to run in a single AppDomain.

**Q-16) Explain role-based security?**

Role-based security is used to implement security measures based on the role assigned to the users in the organization. Then we can authorize users based on their roles in the organization. For example, windows have role-based access like user, administrators, and guests.

**Q-17) What is cross-page posting?**

Whenever we click on a submit button on a page, the data is stored on the same page. But if the data is stored on a different page, it is known as a cross-page posting.

Cross-page posting can be achieved by POSTBACKURL property which causes the postback.

**Q-18) Explain passport authentication?**

During the passport authentication, it first checks the passport authentication cookie, if the cookie is not available the application redirects to the passport sign on page. Passport service then authenticates the details of the user on the sign on page and if they are valid, stores them on the client machine and then redirects the user to the requested page.

**Q-19) What are ASP.NET security controls?**

* <asp: Login>: Provides a login capability that enables the users to enter their credentials.
* <asp: LoginName>: Allows you to display the name of the logged-in user.
* <asp: LoginStatus>: Displays if the user is authenticated or not.
* <asp: LoginView>: provides various login views depending on the template that has been selected.
* <asp: PasswordRecovery>: Emails the users the lost passwords.

**Q-20) List all the templates of the Repeater control?**

* ItemTemplate
* AlternatingItemTemplate
* SeparatorTemplate
* HeaderTemplate
* FooterTemplate

**Q-21) What is MIME?**

MIME stands for multipurpose internet mail extensions, it is the extension of the e-mail protocol which lets users use the protocol to exchange files over the internet.

Servers insert the MIME header at the beginning of the web transmission. Then the clients use this header to select an appropriate ‘player’ for the type of data that the header indicates. Some of these players are built into the web browser.

**Q-22) What are the different types of cookies in ASP.NET?**

* **Session Cookie:**It resides on the client machine for a single session until the user logs out.
* **Persistent Cookie:**Resides on the user machine for a period specified for its expiry. It may be an hour, a month or never.

**Q-23) What is the difference between ExecuteScalar and ExecuteNonQuery?**

|  |  |
| --- | --- |
| **ExecuteScalar** | **ExecuteNonQuery** |
| Returns the output value | Does not return any value |
| Used for fetching a single value | Used to execute insert and update statements |
| Does not return the number of affected rows | Returns the number of affected rows. |

**Q-24) What is the concept of DISPOSE method?**

DISPOSE method belongs to IDisposable interface. It is used to free unmanaged resources like files, network connection etc. It manages and handles this by an instance of the class that implements this interface. Dispose methods must be called explicitly and hence the any object using IDisposable must also implement finalizer to free resources in situations wherein Dispose is not called. Multiple calls to dispose method must be ignored when called once. The objects disposable methods must be called in the order of containment.

**Q-25) What is a delegate in .NET?**

A delegate is a type that encapsulates a reference to a method. Delegate objects can then be passed to code which calls the method according to the method signature, meaning the developer doesn't have to know at compile time which method is being invoked. A delegate can contain references to a single method or multiple methods. Delegates are similar to function pointers in C/C++ and have the advantage of being type-safe.

**Q-26) What is the difference between NameSpace and Assembly?**

A Namespace is a logical naming scheme for types in which a simple type name, such as MyType, is preceded with a dot-separated hierarchical name. Such a naming scheme is completely under control of the developer. The .NET Framework uses a hierarchical naming scheme for grouping types into logical categories of related functionality, such as the ASP.NET application framework, or remoting functionality. Design tools can make use of namespaces to make it easier for developers to browse and reference types in their code.

The concept of a namespace is not related to that of an Assembly. A single assembly may contain types whose hierarchical names have different namespace roots, and a logical namespace root may span multiple assemblies. In the .NET Framework, a Namespace is a logical design-time naming convenience, whereas an Assembly establishes the name scope for types at run time.

**Q-27) What is OOP, and how does it relate to the .NET Framework?**

OOP stands for object-oriented programming. A good answer to this question would point out that OOP languages such as Visual Basic.NET, C# and C++ are the core languages supported by .NET Framework.

As a technique, OOP allows .NET developers to create classes containing methods, properties, fields, events and other logical modules. It also lets developers create modular programs, which they can assemble as applications. OOPs have four basic features: encapsulation, abstraction, polymorphism and inheritance.

**Q-28) What is the difference between a class and an object, and how do these terms relate to each other?**

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.

**Q-29) What is .NET web service?**

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.

**Q-30) Read-only variables and constants have many similarities, but what is at least one way that they differ?**

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.

**Q-31) List down the differences between Public, Static and void keywords?**

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| Public | It is an access specifier which states that the method of a class can be accessed publicly |
| Static | It is a keyword used for declaring a member of a type, specific to that type |
| Void | It states that the method does not return any value |

**Q-32) Illustrate the process of code compilation in C#?**

 There exist four steps in the process of code compilation:

1. Compilation of Source code in managed  code
2. Clubbing the newly created code into assembly
3. Loading the CLR (Common Language Runtime)
4. Execution of assembly through CLR

**Q-33) List down the access modifiers available in C#?**

Following are the access modifiers available for general use:

* **Public-**When an attribute or method is defined as public it can be accessed from any part of code.
* **Private-**A private attribute or method can be accessed from within the class itself.
* **Protected-**When a user defines a method or attribute as protected then it can be accessed only within that class and the one inheriting the class.
* **Internal-**When an attribute or method is defined as internal then it will be accessed from that class at the current assembly position.
* **Protected Internal-**When you define an attribute or method as protected internal, then it’s access restricted to classes within the current project assembly or different types defined by that class

**Q-34) What are the different approaches of passing parameters to a method?**

There are three ways of passing parameters to a method:

**Value Parameters-** Under this method the actual value of an argument is copied to the formal parameter of the function. In, this case the changes made into the formal parameter of the function have no effect on the actual value of the argument.

**Reference Parameters-**This method copies the argument referring to the memory location into the formal parameter. Meaning changes made to the parameter affect the argument.

**Output Parameters-**This method returns more than one value.

**Q-35) Distinguish between finally and finalize blocks?**

The finally block is called after the execution of try and catch blocks, It is used for exception handling whether or not the exception has been caught this block of code gets executed. Generally, this block of code has a cleaner code.

The finalize method is called just before the garbage collection. Main priorities are to perform clean up operation for unmanaged code, it is automatically invoked when an instance is not subsequently called.

**Q-36) Define sealed classes in C#?**

You create sealed classes in situations when you want to restrict the class to be inherited. For doing this sealed modifiers are used. If you forcefully specify a sealed class as a base class then a compilation error occurs.

**Q-37) Define a Partial class?**

A partial class is the only one that essentially splits the definition of a class into multiple classes in either same source code files or multiple source code files. One can create a class definition in multiple files but that is compiled as one class at run-time and when an instance of this class is created, one can access all methods from every source file with the same object. It is indicated by the keyword ‘partial’.

**Q-38) What is the difference between Typed and Untyped Dataset?**

**Typed Dataset:** A typed dataset is derived from the Dataset class and has an associated XML schema, which is created at the time of creation of the dataset.

The XML schema contains information about the dataset structure such as tables, columns, and rows. Data is transferred from a database into a dataset and from the dataset to another component in the XML format.

**Untyped Dataset:** Untyped dataset doesn’t have an XML schema associated with it. Untyped Dataset, the tables, and columns are represented as a collection.

**Q-39) Explain StreamReader/StreamWriter class?**

Streamreader and StreamWriter are classes of namespace.System.IO. Used when we want to read or write charact90, Reader based data, respectively.

Members of StreamReader are:**Close(),Read(),Readline().**

Members of Streamwriter are: **close()**, **write()**, **writeline()**.

**Example program:**

|  |  |
| --- | --- |
|  | Class Testprogram  {  using(StreamReader sr = new StreamReader("C:Readme.txt")  {  // Any code to read//  }  using(StreamWriter sr = ndew StreamWriter("C:Readme.txt")  {  // Any code to write//  }  } |

**Q-40) Define using statement in C#?**

“Using” keyword simply denotes that the particular namespace is being used by the program. For ex- using System, Here System is a namespace. The class console is defined under system, so we can use the Console.Writeline(…) or Readline in our program.

**Q-41) Can you return multiple values from a function in C#?**

Yes! Using output parameters. A return statement can be used for returning only one value from a function. However, using output parameters, you can return two values from a function.

## Q-42) What is a Destructor in C#?

A **Destructor** is used to clean up the memory and free the resources. But in C# this is done by the garbage collector on its own. System.GC.Collect() is called internally for cleaning up. But sometimes it may be necessary to implement destructors manually.

**Q-43) What is the difference between overloading and overriding?**

|  |  |
| --- | --- |
| **Overloading** | **Overriding** |
| **Overloading** is when you have multiple methods in the same scope, with the same name but different signatures. | **Overriding** is a principle that allows you to change the functionality of a method in a child class. |
| **Example:**  public class sample  {  Public void getStuff(int id)  {}  public void getStuff(string name)  {}  } | **Example:**  public class sample1  {  Public virtual void getStuff(int id)  {  //Get stuff default location  }  public class sample2:sample1  {  public override void getStuff(int id)  {  //base.getStuff(id);  //or-Get stuff new location  }  } |

## Q-44) What are dynamic type variables in C#?

You can store any type of value in the dynamic data type variable. Type checking for these types of variables takes place at run-time.

**Q-45) Define a Jagged Array in C#?**

A Jagged array is referred to as an “array of arrays”. It is an array whose elements are arrays, the element of the same can be of different dimensions and sizes. The length of each array index can differ.

**Example:** int[][] jagArray = new int[5][];

**Q-46) Define an Escape Sequence, Name few strings in Escape Sequence?**

An Escape Sequence is denoted by a backslash (). The backslash merely indicates that the character it is following should be interpreted literally or is a special character. An escape sequence is a single character.

Few escape sequences are as follows:

n – newline character

b – backspace

– backlash

‘ – Single quote

” – Double quote

**Q -47) What is Databinding?**

Databinding is the process of binding the data with graphical elements (controls in a window form). After binding the data in a window form, you can navigate through the records with the help of the Binding Navigator Control.

One of the advantages of data binding is, the user does not need to write the codes explicitly, for establishing the connections and creating a data set, this feature will write the necessary ADO.NET code for the user.

**Two types of Databinding:**

* **Simple Data Binding:** It is the process of binding a control to a single data element. **Example:** Binding a textbox or label with a single value.
* **Complex Data Binding:** It is the process of binding a component to display one data element or multiple data elements.

**Example:** GridView or Listbox control binding one data element or more than one records in a table.

**Q-48) Distinguish between Array and Arraylist in C#?**

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| Array uses the vector array for storing the elements. | ArrayList uses the LinkedList to store the elements. |
| Size of the Array must be defined until redim used(vb). | There’s no need for specifying storage size. |
| An array is a specific data type storage. | ArrayList can store everything as an object. |
| Typecasting is not necessary. | Typecasting is necessary. |
| There is no RunTime exception. | There is a RunTime error exception. |
| Elements can’t be inserted or deleted in between. | Elements can be inserted or deleted in between. |

**Q-49) What are the uses of delegates in C#?**

Below are the list of uses of delegates in C#:

* Callback Mechanism
* Asynchronous Processing
* Abstract and Encapsulate method
* Multicasting

**Q-50) What is deep or shallow copy concept in C#?**

* **Shallow Copy** is about copying an object's value type fields into the target object and the object's reference types are copied as references into the target object but not the referenced object itself. It copies the types bit by bit. The result is that both instances are cloned and the original will refer to the same object.
* **Deep Copy** is used to make a complete deep copy of the internal reference types, for this we need to configure the object returned by MemberwiseClone().

**Q-51) What Is The Difference Between Reflection And Dynamic?**

Both Reflection and dynamic are used to operate on an object during run time. But they have some differences:

* + Dynamic uses reflection internally
  + Reflection can invoke both public and private members of an object. But dynamic can only invoke public members of an object

**Q-52) Differentiate Static Polymorphism and Runtime Polymorphism?**

**Static Polymorphism:** The object decides about the function execution during compile time itself. The Process of binding object with function called Static Binding or Early binding. This is achieved by using Function Overloading.

**Runtime Polymorphism:** The object decides about the function execution during run time itself. The Process of binding object with function called Runtime or Late binding. This is achieved by using Function Overriding(Abstract Class + Virtual Function).

**Q-53)** **Why Serialization And Deserialization?**

For example consider, we have a very complex object and we need XML format to show it on HTML page. Then we can create a XML file in the disk, writes all the necessary data on the XML file, and use it for the HTML page. But this is not good approach for large number of users. Extra space is required; anyone can see the XML file which creates security issue. We can overcome it by using XML serialization.

**Q-54) How is Exception Handling implemented in C#?**

Exception handling is done using four keywords in C#:

* **try** – Contains a block of code for which an exception will be checked.
* **catch** – It is a program that catches an exception with the help of exception handler.
* **finally** – It is a block of code written to execute regardless whether an exception is caught or not.
* **Throw** – Throws an exception when a problem occurs.

**Syntax for catching an exception:**

To catch an Exception we make use of try-catch blocks. The catch block has a parameter of the system.Exception type.

**Example:**

|  |
| --- |
| Try  {  GetAllData();  }  catch(Exception ex){  } |

## Q-55) Why to use “finally” block in C#?

**Finally** block will be executed irrespective of exception. So while executing the code in try block when exception is occurred, control is returned to catch block and at last finally block will be executed. So closing connection to database / releasing the file handlers can be kept in finally block.

**Q-56) What are Events?**

Events in C# follow a concept where it consists of a Publisher, Subscriber, Notification and a handler. You can think of an event as nothing but an encapsulated delegate.

**Example:**

|  |  |
| --- | --- |
|  | public Delegate void TestEvent();  public TestEvent TestEvent1; |

**Q-57) Define C# I/O classes? List the commonly used classes?**

C# consists of System.IO namespace which has classes that compute and perform various operations on files like creating, deleting, opening and closing etc.

Few commonly used I/O classes are listed below:

**File-**Helps in manipulating file.  
**StreamWriter-**Generally used for writting characters to a stream.  
**StreamReader-**Generally used for reading characters to a stream.  
**StringWriter-** Used for writing a string buffer.  
**Stringreader-**Used for reading a string buffer.

**Q-58) Explain about generics in C#.NET?**

 Generics are used to make reusable code classes which decrease the code redundancy,

* Increase type safety, performance, and optimization.
* Using Generics one can do a variety of things like create collections.
* To create Generic collection, System.namespace.
* The generic namespace should be used inspite of classes such as ArrayList in the System.
* Generics instigates the usage of a parameterized type.

**Q-59) What are the key events of SqlConnection Class?**

**The two key events of SqlConnection are:**

* **StateChange event:**This event occurred when the state of the Connection changes. The event handler receives an argument (Datatype: StateChangeEventArgs) which contains the data related to that particular event.
* **InfoMessage event:** This event occurred when an info message or Warning is returned from a data source. The event handler receives an argument (Datatype: SqlInfoMessageEventArgs) which contains the data related to that particular event.

**Q -60) What you mean by Filtering of data?**

Filtering of data is done when you need to display only selective records.

**Given below are the  two methods for filtering data:**

* Creating parameterized queries.
* Filtering data using control of a window form.

**Q -61) What is the difference between Connected and Disconnected environments?**

|  |  |
| --- | --- |
| **Connected Environment** | **Disconnected Environment** |
| It requires a constant connection to transfer data between the application and database. | It doesn’t require a constant connection to transfer data between the application and database. |
| Data concurrency is easy to control. | Data concurrency is not easy to control. |
| Data is up-to-date since user is always connected to the database. | Data is not up-to-date since user is always connected to the database. |
| It has scalability and performance issues for the client application. | Data is not up-to-date since user is always connected to the database. |
| It has scalability and performance issues for the client application. | It improves scalability and performance of the client application. |

**Q-62) Explain Synchronous and Asynchronous Operations?**

Synchronization is a way of creating a thread-safe code where only a single thread will access the code in a given time. A synchronous call waits for completion of method and then continous the program flow. Synchronous programming adversely affects the UI operations that normally happens when user tries to perform time-consuming operations since only one thread is used.

In Asynchronous operation, the method call immediately returns allowing the program to perform other operations while the method called completes its share of work in certain circumstances.

**Q-63) What is Thread Pooling?**

A Thread pool is a collection of threads that perform tasks without disturbing the primary thread. Once the task is completed by a thread it returns to the primary thread.

**Q-64) What is ASP.NET?**

ASP.NET is Microsoft's framework to build Web applications. ASP.NET is a part of .NET Framework. ASP.NET and Web Forms are used to build the front end and in the backend, C# language is used. ASP.NET runs on a Web Server, IIS.  
  
**Advantages of ASP.NET:**

**1. Separation of Code from HTML:** To make a clean sweep, with ASP.NET you have the ability to completely separate layout and business logic. This makes it much easier for teams of programmers and designers to collaborate efficiently.

**2. Support for compiled languages:** Developers can use VB.NET and access features such as strong typing and object-oriented programming. Using compiled languages also means that ASP.NET pages do not suffer the performance penalties associated with interpreted code. ASP.NET pages are precompiled to byte-code and Just In Time (JIT) compiled when first requested. Subsequent requests are directed to the fully compiled code, which is cached until the source changes.

**3. Use services provided by the .NET Framework:** The .NET Framework provides class libraries that can be used by your application. Some of the key classes help you with input/output, access to operating system services, data access, or even debugging. We will go into more detail on some of them in this module.

**4. Graphical Development Environment:** Visual Studio .NET provides a very rich development environment for web developers. You can drag and drop controls and set properties the way you do in Visual Basic 6. And you have full IntelliSense support, not only for your code but also for HTML and XML.

**5. State management:** To refer to the problems mentioned before, ASP.NET provides solutions for session and application state management. State information can, for example, be kept in memory or stored in a database. It can be shared across web farms, and state information can be recovered, even if the server fails or the connection breaks down.

**6. Update files while the server is running:** Components of your application can be updated while the server is online and clients are connected. The framework will use the new files as soon as they are copied to the application. Removed or old files that are still in use are kept in memory until the clients have finished.

**7. XML-Based Configuration Files:** Configuration settings in ASP.NET are stored in XML files that you can easily read and edit. You can also easily copy these to another server, along with the other files that comprise your application.

**Q-65) What are the different validators in ASP.NET?**

ASP.NET validation controls define an important role in validating the user input data. Whenever the user gives the input, it must always be validated before sending it across to various layers of an application. If we get the user input with validation, then chances are that we are sending the wrong data. So, validation is a good idea to do whenever we are taking input from the user.  
  
There are the following two types of validation in ASP.NET,

* Client-Side Validation
* Server-Side Validation

**Client-Side Validation**

When validation is done on the client browser, then it is known as Client-Side Validation. We use JavaScript to do the Client-Side Validation.  
  
**Server-Side Validation**  
When validation occurs on the server, then it is known as Server-Side Validation. Server-Side Validation is a secure form of validation. The main advantage of Server-Side Validation is if the user somehow bypasses the Client-Side Validation, we can still catch the problem on server-side.  
  
The following are the Validation Controls in ASP.NET,

* RequiredFieldValidator Control
* CompareValidator Control
* RangeValidator Control
* RegularExpressionValidator Control
* CustomFieldValidator Control
* ValidationSummary

**Q-66) Explain brief about View State?**

View State is the method to preserve the Value of the Page and Controls between round trips. It is a Page-Level State Management technique. View State is turned on by default and normally serializes the data in every control on the page regardless of whether it is actually used during a post-back.

A web application is stateless. That means that a new instance of a page is created every time when we make a request to the server to get the page and after the round trip our page has been lost immediately

**Features of View State**  
These are the main features of view state,

1. Retains the value of the Control after post-back without using a session.
2. Stores the value of Pages and Control Properties defined in the page.
3. Creates a custom View State Provider that lets you store View State Information in a SQL Server Database or in another data store.

**Advantages of View State**

1. Easy to Implement.
2. No server resources are required: The View State is contained in a structure within the page load.
3. Enhanced security features: It can be encoded and compressed or Unicode implementation.

**Q-67) What is Ajax in ASP.NET?**

Ajax stands for Asynchronous JavaScript and XML; in other words Ajax is the combination of various technologies such as a JavaScript, CSS, XHTML, DOM, etc.  
  
AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the entire page.  
  
We can also define Ajax is a combination of client-side technologies that provides asynchronous communication between the user interface and the web server so that partial page rendering occurs instead of a complete page postback.  
  
Ajax is platform-independent; in other words, AJAX is a cross-platform technology that can be used on any Operating System since it is based on XML & JavaScript. It also supports open source implementation of other technology. It partially renders the page to the server instead of the complete page being post back. We use AJAX for developing faster, better and more interactive web applications. AJAX uses an HTTP request between web server & browser.

* With AJAX, when a user clicks a button, you can use JavaScript and DHTML to immediately update the UI, and spawn an asynchronous request to the server to fetch results.
* When the response is generated, you can then use JavaScript and CSS to update your UI accordingly without refreshing the entire page. While this is happening, the form on the user's screen doesn't flash, blink, disappear, or stall.
* The power of AJAX lies in its ability to communicate with the server asynchronously, using a XMLHttpRequest object without requiring a browser refresh.
* Ajax essentially puts JavaScript technology and the XMLHttpRequest object between your Web form and the server.

**Q-68) What are Web Services in ASP.NET?**

A Web Service is a software program that uses XML to exchange information with other software via common internet protocols. In a simple sense, Web Services are a way of interacting with objects over the Internet.  
  
A web service is,

* Language Independent.
* Protocol Independent.
* Platform Independent.
* It assumes a stateless service architecture.
* Scalable (e.g. multiplying two numbers together to an entire customer-relationship management system).
* Programmable (encapsulates a task).
* Based on XML (open, text-based standard).
* Self-describing (metadata for access and use).
* Discoverable (search and locate in registries)- ability of applications and developers to search for and locate desired Web services through registries. This is based on UDDI.

Key Web Service Technologies,

* XML- Describes only data. So, any application that understands XML-regardless of the application's programming language or platform-has the ability to format XML in a variety of ways (well-formed or valid).
* SOAP- Provides a communication mechanism between services and applications.
* WSDL- Offers a uniform method of describing web services to other programs.
* UDDI- Enables the creation of searchable Web services registries.

**Q-69) Can you explain the page life cycle of MVC?**

Below are the processed followed in the sequence -

* App initialization
* Routing
* Instantiate and execute controller
* Locate and invoke controller action
* Instantiate and render view.

**Q-70) Explain Dependency Resolution?**

Dependency Resolver again has been introduced in MVC3 and it is greatly simplified the use of dependency injection in your applications. This turn to be easier and useful for decoupling the application components and making them easier to test and more configurable.

**[Q-71) List the two important objects of ADO.net and also list the namespaces that are commonly used in ADO.net to aid in connection to a database.](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled2)**

Two of the most important objects of ADO.net are the DataReader and DataSet. Some of the commonly used namespaces that can help in connecting to a database are:

* System.Data.OleDb namespace
* System.Data.SQLClient namespace
* System.Data namespace

**[Q-72) List some of the common data providers for ADO.net framework.](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled3)**

Some of the common data providers for ADO.net framework include:

* The .net framework data provider for Oracle
* The .net framework data provider for OLE DB
* The .net framework data provider for ODBC
* The .net framework data provider for SQL server

**[Q-73) What do you mean by Object Pooling and Connection Pooling?](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled5)**

**Object pooling** in terms of ADO.net framework is a repository of objects stored in memory cells that are can be used later. The benefit of using object pooling is that it reduced a load of the creation of new objects each time when required. When the user required any object, the manager of the object pool considers a request and processes it accordingly.

**Connection pooling** is a set of database connections that are used or reused when required after processing a request. The merit of utilizing a connection pooling database is that it increases the performance of execution of database commands. It also decreases the time and effort taken to carry out operations.

**[Q-74) Differentiate between DataSet and DataReader?](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled7)**

|  |  |
| --- | --- |
| DataSet | DataReader |
| Works on loops | Works on forward only |
| Has disconnected recordsets | Has ample of connected recordsets |
| Can carry out both addition and deletion | Can perform only read-only activities |
| Can store XML data | There is no facility for storage of XML data |
| There are multiple tables involved | Only requires a single table |
| There must be a relationship between the tables | No relationship between the tables required |
| Requires more memory space | Requires only less memory space |

**[Q-75) Enlist all the components of an ADO.net framework?](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled8)**

The components of an ADO.net framework include:

* **Connection object:** It is a representation of the connection to the databases
* **DataReader:** It performs only forward and read-only activities on the record set
* **DataAdapter:** It acts as a bridge between the database and dataset
* **Command object:** It is used to execute and store procedures and commands that are required on a database
* **ExecuteNonQuery:** This component is used to execute commands but it does not return any value
* **ExecuteScalar:** Performs execution and returns of single value
* **ExecuteReader:** Performs execution and returns result sets

**[Q-76) What are some of the commonly used commands in SQLCommand?](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled11)**

Some of the commonly used commands in SQL command in ADO.net framework include:

* **Cancel:** Used when the user wants to cancel a request.
* **ExecuteScalar:** Used to execute and return a set of single values.
* **ResetCommandTimeout:** Used in order to reset any timeout property.
* **ExecuteReader:** Used for execution and returning data back into the DataReader.
* **ExecuteNotQuery:** This command is used by the user in order to execute and does not require any result set in return.
* **ExecuteXMLReader:** Used when the user requires the execution and requires the resultant data in the XMLDataReader object.

**[Q-77) What is the importance of closing an ADO.net application?](https://www.onlineinterviewquestions.com/ado-net-interview-questions/" \l "collapseUnfiled13)**

Closing an application in the ADO.net framework is essential because it can negatively affect the reliability and scalability of the network. In case of open connection, there is always the possibility of attack from various sources of other open connections.

**Q-78) What is the difference between ADO and ADO.NET?**

|  |  |
| --- | --- |
| ADO | ADO.NET |
| ADO has one main object that is used to reference data, called the RecordSet object. | ADO.NET provides objects that allow you to access data in various ways. The DataSetobject allows you to store the relational model of your [database](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx). MARS (Multiple Active Result Sets) is implemented in ADO.NET. |
| You can only work on connected manner. This means that when you access data, such as viewing and updating data, it is real-time, with a connection being used all the time. This is barring, of course, you programming special routines to pull all your data into temporary tables.  In connected model you always get refreshed data. | ADO.NET uses data in a disconnected fashion. When you access data, ADO.NET makes a copy of the data using XML. ADO.NET only holds the connection open long enough to either pull down the data or to make any requested updates. This makes ADO.NET efficient to use for [Web applications.](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx) It's also decent for desktop applications.  You can work on connected and disconnected manner.br> In disconnected model you will get old data as you are editing it. Outlook is an example of disconnected model. We work on offline object model and when connection is required it is connected.  Connected object can be used on disconnected object. |
| Whereas ADO allows you to persist records in XML format | ADO.NET allows you to manipulate your data using XML as the primary means. This is nice when you are working with other business applications and also helps when you are working with firewalls because data is passed as HTML and XML. |
| ADO allows you to create client-side cursors only. | ADO.NET gives you the choice of either using client-side or server-side cursors. In ADO.NET, classes actually handle the work of cursors. The developer has the freedom of choice in [internet](http://www.c-sharpcorner.com/UploadFile/puranindia/what-is-ado-net/WhatisADONET.aspx) development, for creating efficient applications. |

**Q-79) Which properties are used to bind a DataGridView control?**

The DataSource property and the DataMember property are used to bind a DataGridView control.

**Q-80) What is the basic difference between ASP and ASP.NET?**

The basic difference between ASP and ASP.NET is that ASP is interpreted; whereas, ASP.NET is compiled. This implies that since ASP uses VBScript; therefore, when an ASP page is executed, it is interpreted. On the other hand, ASP.NET uses .NET languages, such as C# and VB.NET, which are compiled to Microsoft Intermediate Language (MSIL).