- ➔ What is .Net?
  .Net is a developer platform to build a variety of applications for web, mobile, desktop, and loT. It supports languages like C#, F#, VB, J#, C++, etc., for building the applications.

- ➔ What are the components of .Net?
  - Common Language run-time (CLR) – it is provides a secure execution environment for applications, whenever an application is written in C# is complied the code is converted into an intermediate language. After this the code is targeted to CLR which performs several operations: memory management, security checks, loading assemblies, and thread management.
  - Application Domain (AppDomain) - is a logically isolated container inside which the .NET code running in one application cannot adversely affect other, unrelated applications. It provides an isolation boundary for security, reliability, versioning, and for unloading assemblies and are typically created by runtime hosts, which are responsible for bootstrapping the common language runtime before an application is run.
  - Common Type System (CTS) - It follows certain rules according to which a data type should be declared and used in the program code. CTS describes the data types that are going to be used in the application, it helps in calling the data type declared in one program language by other programming languages.
  - .Net Class Library
  - .Net Framework- The the .Net framework is a software development platform that supports an object-oriented approach. It offers services, like memory management, networking, security, and type safety. It supports various languages like C#, VB, Cobol, Perl, .Net, etc. and has a wide variety of tools and functionalities like class, library, and APIs that are used to build, deploy, and run web services and different applications.
  - Profiling

- ➔ Framework Class Library (FCL) - is a collection of reusable types, including classes, interfaces, and data types included in the .NET Framework. It is used for developing a wide variety of applications, as it provides access to system functionality.

- ➔ Base Class Library (BCL) - comprises classes, interface, and value types. It is the foundation for building .NET Framework applications, components, and controls. It encapsulates a huge number of common functions and make them easily available for the developers and provides functionality like threading, input/output, security, diagnostics, resources, globalization, namespaces that are used frequently (i.e. System….), etc.

- ➔ Common Language Specification (CLS) - helps the developers to use the components that are inter-language compatible with certain rules that come with CLS, it then helps in reusing the code in the other .Net compatible languages.

- ➔ Just In Time (JIT) compiler - is used to convert the intermediate code into the native language during execution. It is a compiler in CLR responsible for the execution of .NET programs of different languages by converting them into machine code. It speeds up the code execution and supports multiple platforms.
  - There are 3 types of JIT compilers:
    - Pre-JIT Compiler - It compiles all the source code into the machine code in a single compilation cycle, i.e. at the application deployment time
    - Normal JIT Compiler - The source code methods required at run-time are compiled into

machine code and stored in the cache to be called later.
        - Econo JIT Compiler - The methods required only at run-time are compiled using this compiler and they are not stored for future use.

- **LINQ** (Language Integrated Query) was introduced in 2008. It is a set of features that extend query capabilities to the .Net framework language syntax by adding sets of new standard query operators that allows data manipulation regardless of the data source. Supported data sources are: .NET Framework collections, SQL Server databases, ADO.NET Datasets, XML documents, and any collection of objects that support IEnumerable or the generic IEnumerable<T> interface, in both C# and VB. In short, LINQ bridges the gap between the world of objects and the world of data.

- In LINQ - Deferred execution means that the query is not executed at the time it is specified. This is accomplished by assigning the query to a variable, the variable stores the query definition but the query is not executed until the query variable is iterated over. ex.

```
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)
 }
```

- In LINQ – Immediate execution of a query can be useful if the database is being updated frequently, and it’s 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. Example:

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

- Microsoft Intermediate Language (MSIL) provides instructions for calling methods, storing and initializing values, memory handling, exception handling, etc. All the .Net codes are first compiled to Intermediate Language (IL).

- Assembly – the simple collection of all the logical units present, can be considered as a collection of executable and dynamic link library (DLL) files. Logical units are entities that are required to build an application and later deploy the application using the .Net framework
    - Different parts of the Assembly are:
        - Manifest (aka – assembly metadata): it has the information about the version of the assembly
        - Type Metadata: contains the binary information of the program
        - MSIL
        - Resources: list of related files
    - There are two types of assemblies: Private and Shared

        - Private assembly is accessible only to the application, it is installed in the installation directory of the application
        - Shared assembly can be shared by multiple applications, it is installed in the Global Assembly Cache (GAC) which is a folder specifically designed to be shared by all applications executed on a system

➔ EXE and DLL are assembly executable modules.

EXE - executable file that runs the application for which it is designed. When we build an application, an exe file is generated. The assemblies are loaded directly when we run an exe, an exe file cannot be shared with other applications.
DLL - stands for dynamic link library, it consists of code that needs to be hidden. The code is encapsulated in this library, an application can have many DLLs and can also be shared with other applications.

➔ What is the difference between Response.Redirect and Server.Transfer?
- Response.Redirect redirects the user's browser to another page or site, the history of the browser is updated to reflect the new address. It also preforms a trip back to the client where the client's browser is redirected to the new page
- Server.Transfer transfers from one page to the other without making any round trip back to the client's browser and the history is not updated.

➔ 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.
- FindControl method can be used to get the values that are posted on this page to which the page has been posted.

➔ Differences between Managed code and Unmanaged code:
- 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, and not by the operating system itself. CLR and managed code offers developers few benefits, like garbage collection, type checking and exceptions handling.
- 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++.

➔ How does managed code execute in .Net framework?
- Four main steps that are included in the execution of managed code:
    - Choose a compiler depending on the language of the code written
    - Convert the code into Intermediate Language (IL) using a compiler
    - IL is targeted to CLR which converts it to native code using JIT
    - Native code is executed using the .Net runtime

➔ Code Access Security (CAS) – is part of a security model that prevents unauthorized access to the resources, it enables users to set permissions for the code that CLR then executes depending on the permissions. CAS can only be used for managed code. If an assembly uses CAS it is treated as only partially trusted although it goes through checks each time an assembly tries to access the resources

- ➔ Model View Controller (MVC) – is an architecture model that is used to build .Net applications
  - Model - logical part of any application that handles the object storage and retrieval from the databases for an application
  - View – handles the user interface (UI) of an application, gets the information from the models for their display
  - Controller – handles the user interactions, figure out the responses for the user input and render the View that is required for the user interaction

- ➔ State management is used to constantly monitor and maintain the state of objects in run-time (a web page or a controller is considered an object) There are two types of state management:
  - **Client-side**: used to store information on the client's machine and is formed mostly of reusable and simple objects
  - **Server-side**: stores the information on the server and makes it easier to manage and preserve the information on the server

- ➔ different validators in ASP.NET?
  - **Client-side** validation – When the validation takes place on the client-side browser, it is called client-side validation. Usually, JavaScript is used for client-side validation.
  - **Server-side** validation – When the validation takes place on the server then it is called server-side validation. Server-side validation is considered as a secure form of validation because even if the user bypasses the client-side validation we can still catch it in server-side validation.

- ➔ Caching is a term used when the data has to be temporarily stored in the memory so that an application can access it quickly rather than looking for it in a hard drive. This speeds up the execution to an exponential pace and helps massively in terms of performance.
  - There are three types of caching:
    - Data caching
    - Page caching
    - Fragment caching

- ➔ There are five main types of constructor classes in C# as listed below:
  - Copy constructor
  - Default constructor
  - Parameterized constructor
  - Private constructor
  - Static constructor

- ➔ A Session object stores information and variables about a user and retains it through the session There are numerous advantages of making use of a session are as mentioned below:
  - It is used to store user data across the span of an application.
  - It is very easy to implement and store any sort of object in the program.
  - Individual entities of user data can be stored separately if required.
  - The session is secure, and objects get stored on the runtime server. Yes, it is possible to

manually set a session’s out time. It can easily be done by manipulating the web.config file.

- ➔ Garbage collection is a process that is used to maintain various aspects of memory to prevent memory leaks during program execution. An entity called the garbage collector is used to allocate and de-allocate memory as and when required by an application. This is done by performing checks on the references of variables and objects used by the application. If an object is no longer required by the application, the memory is de-allocated and freed up. The three generations of garbage collection are:
  - **Gen 0**: Stores short-lived objects
  - **Gen 1**: Objects moved from Gen 0 live here
  - **Gen 2**: Stores long-lived objects and Gen 1 objects

- ➔ There are two types of memories present in .NET Stack and Heap.
  - Stack are stored value types used for static memory allocation (types inherited from System.ValueType 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, is used for dynamic memory allocation, is responsible for keeping track of the data, or more precise objects, are stored reference types (types inherited from System.Object).

- ➔ Differences between Interface and Abstract Class
  - An interface 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. Used to declare properties, methods, events, and fields.
  - Neither interfaces nor abstract classes can be instantiated.

- ➔ Differences between Class and Object:
  - class is the definition of an object, it is the template or blueprint for the object
  - class describes all the methods, properties, etc
  - object is an instance of a class, a class does not become an object unless instantiated
  - an object is used to access those properties from the class

- ➔ Differences between constants and read-only variables:
  - 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 initialized at the time of declaration or in a constructor

- ➔ Differences between function and stored procedure
  - Function
    - function must return a single value, it can only have the input parameter
    - exception handling is not possible using a try-catch block
    - a stored procedure cannot be called from a function

- Stored Procedure
  - always used to perform a specific task
  - can have both input and output parameters
  - exception handling can be done using a try-catch block
  - a function can be called from a procedure

- ➔ Differences between Custom control and user control
  - Custom control – derives from control, dynamic layout, defines a single control, has full toolbox support, and is loosely coupled
  - User control – dervies from UserControl, static layout, defines a set of concepts, cannot be added to the toolbox, and is tightly coupled

- ➔ Explain localization and globalization:
  - Localization- means changing the already globalized application to a specific language or culture. Microsoft.Extensions.Localization is used to localize the application content.
  - Globalization – is the process of developing applications to support multiple languages. Existing applications can also be converted to support multiple languages

- ➔ A delegate in .NET works similarly to that of a function pointer of other programming languages. It provides a way to encapsulate the reference of a method in an object. A delegate object can be easily passed to a program after this, and then a method, which was referenced earlier, can be called. Custom events can also be created in the class using a delegate. We can use a delegate to create custom event within a class. For example,

```
public delegate void FooDelegate() ;
class FooClass
{
   // custom event
   public event FooDelegate FooEvent;
}
FooClass FooObj = new FooClass()
FooObj.FooEvent += new FooDelegate();
```

- ➔ What are methods provided to System.Object's deriving class types?
  - System.Object is the parent class of all .Net classes (implicit, explicit, or user-created) derive from the System.Object class.
  - System.Object provides the following important methods, among others:
    - ToString—Returns a string that represents the current object
    - both overrides of Equals(object), Equals(object, object)
    - GetHashCode
    - Finalize
    - GetType
    - ReferenceEquals
    - MemberwiseClone
  - Most of these methods provide the basic implementation required of any type that a developer will work with in the .NET stack.

- ➔ What is HTTP Handler?
    - ○ Every request into an ASP.NET application is handled by a specialized component called HTTP handler. It is the most important component for handling ASP.NET application requests. It uses different handlers to serve different files. The handler for web page creates the page and control objects, runs your code and then renders the final HTML.
    - ○ Following are the default HTTP handlers for ASP.NET:
        - ▪ Page Handler(.aspx): Handles web pages
        - ▪ User Control Handler(.ascx): It handles web user control pages
        - ▪ Web Service Handler(.asmx): Handles web service pages
        - ▪ Trace Handler(trace.axd): It handles trace functionality

- ➔ Multipurpose internet mail extensions (MIME), 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.
    - ○ The code to send an email from an ASP.NET application is:

```
mail message = new mail();
message.From = "abc@gmail.com";
message.To = "xyz@gmail.com";
message.Subject = "Test";
message.Body = "hello";

SmtpMail.SmtpServer = "localhost";
SmtpMail.Send(message);
```

- ➔ What are the event handlers that we have for the Global.asax file?

•Application Events:

•Application_Start, Application_End, Application_AcquireRequestState,
•Application_AuthenticateRequest, Application_AuthorizeRequest,
•Application_BeginRequest, Application_Disposed, Application_EndRequest,
•Application_Error, Application_PostRequestHandlerExecute,
•Application_PreRequestHandlerExecute, Application_PreSendRequestContent,
•Application_PreSendRequestHeaders, Application_ReleaseRequestState,
•Application_ResolveRequestCache, Application_UpdateRequestCache

•Session Events:

•Session_Start, Session_End

- ➔ If we want to set the user-defined values for the whole applications, we can use the appSettings block in the web.config file. For example the code below uses the ConnectionString throughout the project for the database connection:

```
<em><configuration>
<appsettings>
<add key= "ConnectionString" value="server=local; pwd=password; database=default" />
</appSettings></em>
```

- ➔ 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.

➔ There are five security controls present in ASP.NET as shown below:
  - <asp: PasswordRecovery>: Used to send an email to a user upon performing a password reset operation
  - <asp: Login>: Gives the provisions of login controls with ID and password fields for users to login via credentials
  - <asp: LoginName>: Used to display the name of the user who has logged into the system
  - <asp: LoginStatus>: Used to denote the authentication flag of the user who has logged in
  - <asp: LoginView>: Used to provide a variety of views based on themes upon user login

➔ 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.

➔ 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

➔ For loop and while loop differences
  - 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.

➔ 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.

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

```
int  i = 13 ;
```

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

➔ What are the different types of cookies in ASP.NET?
  - **Session Cookie**: 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.