

*Sri*  
•NET

# Interview Questions

*Authored by:*

Nagaraju B

&

Kogent Learning Solutions Inc.

*Published by:*



$$RS = \frac{P_o}{2}$$

## Table of Contents

Introduction .....	v
Chapter 1: .NET Framework.....	1
Chapter 2: Programming Concepts.....	15
Chapter 3: Object-Oriented Programming.....	27
Chapter 4: Windows Control.....	39
Chapter 5: ADO.NET .....	67
Chapter 6: Language-Integrated Query .....	79
Chapter 7: Dynamic Programming .....	87
Chapter 8: Extensible Markup Language (XML) .....	91
Chapter 9: ASP.NET .....	99
Chapter 10: Windows and Web Services.....	129
Chapter 11: Windows Workflow Foundation.....	141
Chapter 12: Windows Presentation Foundation and Silverlight .....	147
Chapter 13: Windows Communication Foundation 4.0 .....	165
Chapter 14: ASP.NET AJAX.....	175
Chapter 15: Deployment .....	185
Chapter 16: .NET Assemblies.....	191
Chapter 17: Remoting and Reflection .....	197
Chapter 18: ASP.NET MVC Framework .....	211
Chapter 19: Cloud Computing .....	217
Question Index .....	225
Glossary .....	255

**Limits of Liability/disclaimer of Warranty:** The author and publisher have used their best efforts in preparing this book. The author make no representation or warranties with respect to the accuracy or completeness of the contents of this book, and specifically disclaim any implied warranties of merchantability or fitness of any particular purpose. There are no warranties which extend beyond the descriptions contained in this paragraph. No warranty may be created or extended by sales representatives or written sales materials. The accuracy and completeness of the information provided herein and the opinions stated herein are not guaranteed or warranted to produce any particulars results, and the advice and strategies contained herein may not be suitable for every individual. Neither Dreamtech Press nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

**Trademarks:** All brand names and product names used in this book are trademarks, registered trademarks, or trade names of their respective holders. Dreamtech Press is not associated with any product or vendor mentioned in this book.

**ISBN:** 978-81-7722-733-8

**Edition:** 2011

**Printed at:** ArEmm International, Delhi

## Introduction

Thanks for buying this book! This book is specially prepared for students and professionals who intend to attend interview calls on .NET platform. Generally, while preparing for an interview, you struggle to revise important topics and browse through only some useful concepts, which you feel might be of importance from the interview perspective. Such a frenetic search consumes a lot of time and energy, as you have to scan huge volumes of resources to filter out the required information.

This book provides you specific information in a question answer format that you might require to prepare for an interview on .NET platform. The book has been prepared to ensure an easy to understand and self-explanatory presentation. It's equally excellent for Java professionals thinking of a career change, as the book gives a quick understanding of .NET technology and its in-depth features. The topics covered are based on .NET Framework 4.0, which is the latest version of .NET Framework. The book is divided into the following 19 chapters:

- *Chapter 1 —.NET Framework*, introduces .NET Framework and also discusses few of its benefits. In addition, the chapter explores the basic concepts of .NET Framework, such as Common Language Runtime (CLR), Assembly, Microsoft Intermediate Language (MSIL), Just-In-Time compiler, Portable-Executable (PE) files, garbage collection, and Global Assembly Cache.
- *Chapter 2 —Programming Concepts*, covers programming concepts such as boxing, unboxing, constants, variables, namespaces, while loop, for loop, access modifiers, sub procedures, and functions.
- *Chapter 3 —Object-Oriented Programming* (OOPS), introduces the concept of Object-Oriented Programming (OOPS), such as classes and objects. It describes the four main principles of OOPS - Encapsulation, Inheritance, Polymorphism, and Abstraction. This chapter also explores the concept of structures, abstract classes, methods, and interfaces.
- *Chapter 4 —Windows Controls*, covers Windows Forms controls and describes how to manipulate their behaviors by developing code in Visual Basic and C# programming languages.

- *Chapter 5*—ADO.NET, introduces the features and architecture of ADO.NET. In addition, it covers data providers, dataset, DataReader object, ADO.NET related namespaces, and ADO.NET Entity Framework.
- *Chapter 6*—Language-Integrated Query, covers the basic LINQ queries, the standard query operators in LINQ, anonymous types, lambda expressions, and extension methods. It also covers the new concepts introduced in LINQ such as the Zip operator and PLINQ.
- *Chapter 7*—Dynamic Programming, explores the concept of runtime environment that allows you to use dynamic languages, such as PHP and JavaScript, in your applications.
- *Chapter 8*—Extensible Markup Language, explores the concepts of XML, XML-DOM, XML Schema, DTD, parsers, XSLT.
- *Chapter 9*—ASP.NET, describes ASP.NET Web applications, Web hosting, Web application architecture, and ASP.NET engine.
- *Chapter 10*—Windows and Web Services, covers Windows services, Web services, SOAP protocol, WSDL, describes the use of Web Services in distributed computing architecture.
- *Chapter 11*—Windows Workflow Foundation, includes description of components and types of Windows workflow Foundation. This chapter also covers the differences between flowchart and procedural workflows and guides you with the implementation of conditions and rules in workflows. It includes the various new features introduced in WF 4.0, such as bookmarks.
- *Chapter 12*—Windows Presentation Foundation and Silverlight, covers the advantages of using WPF over Windows Forms. It also includes the architecture and programming model and important classes of WPF. It covers the new and improved features of WPF 4.0. In addition, it describes some important concepts of Silverlight 4.
- *Chapter 13*—Windows Communication Foundation 4.0, introduces WCF and describes the benefits of WCF, WCF layers, and WCF endpoints. In addition to this, it also covers the new RoutingService class and the throttling feature introduced in WF 4.0.
- *Chapter 14*—ASP.NET AJAX, covers the advantages of an asynchronous postback over the traditional postback. This chapter also familiarizes you with the technologies used by ASP.NET AJAX 4.0.
- *Chapter 15*—Deployment, describes the process to deploy Windows as well as Web applications.
- *Chapter 16*—.NET Assemblies, covers assemblies and their different components. Further, the chapter also describes different assembly types, and assembly signing.

*Chapter 17*—Remoting and Reflection, includes the concept of remoting, serialization, binary formatters, and SOAP formatter, in addition to reflection.

*Chapter 18*—ASP.NET MVC Framework, includes the MVC Framework which is an add-on to the existing ASP.NET architecture. It includes the various changes introduced in the new ASP.NET MVC Framework, which is included with .NET Framework 4.

*Chapter 19*—Cloud Computing, includes the various concepts of cloud, which is the computing which is completely based on the Internet. The basic characteristics of cloud computing, the cloud architecture, and the different models of cloud service are included in the chapter.

# 1

## .NET Framework

### 1. What is .NET Framework?

- .NET Framework is a complete environment that allows developers to a develop, run, and deploy the following applications:
  - Console applications
  - Windows Forms applications
  - Windows Presentation Foundation (WPF) applications
  - Web applications (ASP.NET applications)
  - Web services
  - Windows services
  - Service-oriented applications using Windows Communication Foundation (WCF)
  - Workflow-enabled applications using Windows Workflow Foundation (WF)

.NET Framework also enables a developer to create sharable components to be used in distributed computing architecture. .NET Framework supports the object-oriented programming model for multiple languages, such as Visual Basic, Visual C#, and Visual C++. .NET Framework supports multiple programming languages in a manner that allows language interoperability. This implies that each language can use the code written in some other language.

### 2. Explain the different versions of .NET.

- The different versions of .NET Framework are as follows:
  - .NET Framework 1.0 —Refers to the first version of .NET Framework that was released by Microsoft on February 13, 2002. It is a part of Visual Studio .NET 2002, which is the first version of Visual Studio .NET.
  - .NET Framework 1.1 —Refers to the first major upgrade of .NET Framework, which was released on April 3, 2003. It is a part of Visual Studio .NET 2003, which is the second version of Visual

Studio .NET. In contrast to .NET Framework 1.0, .NET Framework 1.1 has in-built support for mobile ASP.NET controls, Open Database Connectivity (ODBC), and Oracle databases. It also provides support for Internet Protocol version 6 (IPv6).

- .NET Framework 2.0—Refers to the second major upgrade of .NET Framework, which was released on January 22, 2006. It is a part of Visual Studio 2005 and Microsoft SQL Server 2005.

.NET Framework 2.0 was the latest version of .NET Framework that has support for Windows 2000. .NET Framework 2.0 has many changes and enhancements as compared to .NET Framework 1.1. It has a number of Application Programming Interface (API) changes. It contains many new ASP.NET Web controls and data controls. It also contains new personalization features for ASP.NET, for example, support for themes, skins, and webparts.

.NET Framework 3.0—Refers to the third major upgrade of .NET Framework, which was released on November 21, 2006. It contains a set of managed code APIs that form an integral part of Windows Vista and Windows Server 2008. .NET Framework 3.0 uses the same version of CLR that was incorporated with .NET Framework 2.0. .NET Framework 3.0 includes the following four major components:

- Windows Presentation Foundation
- Windows Communication Foundation
- Windows Workflow Foundation
- Windows CardSpace (WCS)

.NET Framework 3.5—Refers to the fourth major upgrade of .NET Framework, which was released on November 19, 2007. Similar to .NET Framework 3.0, .NET Framework 3.5 also uses the CLR version 2.0. .NET Framework 3.5 also installs .NET Framework 2.0 Service Pack (SP) 1, .NET Framework 2.0 SP2 (with 3.5 SP1), and .NET Framework 3.0 SP1, which includes methods and properties that are required for the .NET Framework 3.5 features, such as Language-Integrated Query (LINQ). In addition to LINQ, .NET Framework 3.5 includes many other new features, such as extension methods, lambda expressions, anonymous types, and built-in support for ASP.NET AJAX.

- .NET Framework 4.0—Refers to the latest version of .NET Framework, which was released on April 12, 2010. .NET 4.0 uses the CLR version 4. .NET 4.0 includes PLINQ (Parallel LINQ), which is the parallel implementation of the LINQ engine and Task Parallel Library. It supports Code Contracts and provides

new types to work with arbitrary-precision arithmetic (`System.Numerics.BigInteger`) and complex numbers (`System.Numerics.Complex`).

### 3. What are the main components of .NET Framework?

■ .NET Framework provides enormous advantages to software developers in comparison to the advantages provided by other platforms. Microsoft has united various modern as well as existing technologies of software development in .NET Framework. These technologies are used by developers to develop highly efficient applications for modern as well as future business needs. The following are the key components of .NET Framework:

- .NET Framework Class Library—Contains a rich collection of classes that are readily available for developers. Microsoft has developed these classes to fulfill various tasks of applications, such as working with files and other data storages, performing input-output operations, and drawing graphics. The classes in the Framework class library (FCL) are logically grouped under various namespaces, such as `System`, Microsoft.
- Common Language Runtime—Facilitates the interoperability between different .NET languages, such as C#, Visual Basic, or Visual C++, by providing a common environment for the execution of code written in any of these languages. When you run a .NET application, the language compiler first compiles the source code into an intermediate code called Microsoft Intermediate Language (MSIL) code. The MSIL code is then used by the Just-In-Time (JIT) compiler to convert the MSIL code into the native machine code, which is the final executable code. CLR also manages the allocation and de-allocation of system resources, such as memory for execution process of .NET programs.
- Dynamic Language Runtimes (DLR)—Enables you to run dynamic languages on .NET Framework by adding some additional services to the CLR.
- Application Domains—Provides isolation between applications. In other words, application domains are used to isolate the processes of different applications and can be defined by .NET Framework.
- Runtime Host—Performs the following tasks:
  1. Loads the runtime environment into execution process of a .NET application
  2. Creates an application domain in the .NET process

**6. Describe the roles of CLR in .NET Framework.**

- CLR provides an environment to execute .NET applications on target machines. CLR is also a common runtime environment for all .NET code irrespective of their programming language, as the compilers of respective language in .NET Framework convert every source code into a common language known as MSIL or IL (Intermediate Language).

CLR also provides various services to execute processes, such as memory management service and security services. CLR performs various tasks to manage the execution process of .NET applications. The responsibilities of CLR are listed as follows:

- **Automatic memory management**—Specifies that CLR invokes various built-in functions of .NET Framework to allocate and de-allocate the memory of .NET objects. Therefore, programmers need not write the code to explicitly allocate and de-allocate memory to programs.

- **Garbage Collection**—Prevents memory leaks during program execution. The Garbage collector of CLR automatically determines the best time to free the memory, which is reserved by an object for execution.

□ **Code Access Security**—Imposes restrictions and security during execution of programs. CLR uses security objects to manage access to code during execution of .NET applications. CLR allows an executing code to perform only those tasks for which it has permission. CLR also checks for user's permissions by using authentication and configuration files of .NET applications.

- **Code verification**—Specifies that CLR enforces type safety and prevents a source code from performing illegal operations.

- **JIT compilation of .NET code**—Loads MSIL code on target machine for execution. CLR uses the services of the Just-in-time (JIT) compiler to execute an application.

**7. What is the role of the JIT compiler in .NET Framework?**

- The JIT compiler is an important element of CLR, which loads MSIL on target machines for execution. The MSIL is stored in .NET assemblies after the developer has compiled the code written in any .NET-compliant programming language, such as Visual Basic and C#.

JIT compiler translates the MSIL code of an assembly and uses the CPU architecture of the target machine to execute a .NET application. It also stores the resulting native code so that it is accessible for subsequent calls. If a code executing on a target machine calls a non-native method, the JIT compiler converts the MSIL of that method into native code. JIT compiler also enforces type-safety in

- runtime environment of .NET Framework. It checks for the values that are passed to parameters of any method. For example, the JIT compiler detects any event, if a user tries to assign a 32-bit value to a parameter that can only accept 8-bit value.

**8. What is Microsoft Intermediate Language (MSIL)?**

- The .NET Framework is shipped with compilers of all .NET programming languages to develop programs. There are separate compilers for the Visual Basic, C#, and Visual C++ programming languages in .NET Framework. Each .NET compiler produces an intermediate code after compiling the source code. The intermediate code is common for all languages and is understandable only to .NET environment. This intermediate code is known as MSIL.

**9. What is Common Language Specification (CLS)?**

- CLS is a set of basic rules, which must be followed by each .NET language to be a .NET-compliant language. It enables interoperability between two .NET-compliant languages. CLS is a subset of CTS; therefore, the languages supported by CLS can use each other's class libraries similar to their own. Application programming interfaces (APIs), which are designed by following the rules defined in CLS can be used by all .NET-compliant languages.

**10. What is Common Type System (CTS)?**

- CTS is the component of CLR through which .NET Framework provides support for multiple languages because it contains a type system that is common across all the languages. Two CTS-compliant languages do not require type conversion when calling the code written in one language from within the code written in another language. CTS provide a base set of data types for all the languages supported by .NET Framework. This means that the size of integer and long variables is same across all .NET-compliant programming languages. However, each language uses aliases for the base data types provided by CTS. For example, CTS uses the data type System.Int32 to represent a 4 byte integer value; however, Visual Basic uses the alias Integer for the same; whereas, C# uses the alias int. This is done for the sake of clarity and simplicity.

**11. Differentiate between managed and unmanaged code?**

- Managed code is the code that is executed directly by the CLR instead of the operating system. The code compiler first compiles the managed code to intermediate language (IL) code, also called as MSIL code. This code doesn't depend on machine configurations and can be executed on different machines.

3. Loads the code which is developed by programmers in the application domain

- *Common Type System*—Maintains data integrity across the code written in different .NET-compliant programming languages. CTS ensures that objects of the programs that are written in different programming languages can communicate with each other to share data. CTS prevents data loss when a type in one language transfers data to its equivalent type in other language. For example, CTS ensures that data is not lost while transferring an integer variable of Visual Basic code to an integer variable of C# code. It identifies the types supported by CLR.
- *Metadata and Self-Describing Components*—Specifies that .NET Framework uses metadata of a code or an assembly to exchange information with other code. Metadata can also be used by .NET runtime to load the resources, such as classes and .dll files, to execute a .NET application. Metadata-based declarative information allows various modules and assemblies to communicate with each other irrespective of their hardware and vendors.
- *Cross-Language Interoperability*—Explains how managed objects created in different programming languages can interact with one another.
- *.NET Framework Security*—Specifies that .NET Framework consists of technologies and tools that can be used by developers to protect the resources and code from unauthorized users. .NET Framework tools, such as Caspol.exe, are used to define the permissions to access .NET code; whereas, configuration files, such as .config files, are used to apply various modes of user authentication. A new **Security Transport Model** is also introduced in .NET Framework 4.0.
- *Profiling*—Enables you to use the profilers dynamically with the running process. After a profiler performs all the requested profiling tasks, you can detach that profiler.
- *Side-by-Side Execution*—Enables an application to work with multiple versions of CLR simultaneously, that is, in the same process.

#### 4. List the new features added in .NET Framework 4.0.

- The following are the new features of .NET Framework 4.0:
  - *Improved Application Compatibility and Deployment Support*—Helps you to deploy all the applications and components build with earlier versions on .NET 4.0.

#### 5. What is difference between System.String and System.StringBuilder classes?

- String and StringBuilder classes are used to store string values but the difference in them is that String is immutable (read only) by nature, because a value once assigned to a String object cannot be changed after its creation. When the value in the String object is modified, a new object is created, in memory, with a new value assigned to the String object. On the other hand, the StringBuilder class is mutable, as it occupies the same space even if you change the value. The StringBuilder class is more efficient where you have to perform a large amount of string manipulation.

- *Dynamic Language Runtime*—Supports dynamic languages to run on .NET.
- *Managed Extensibility Framework*—Facilitates the addition of external extensions to your application.
- *Parallel Programming framework*—Supports better use of multi-core and multi-processor systems by using PLINQ, Task Parallel Library, and Coordination Data Structures (CDS). CDS helps you in synchronizing and co-ordinating the execution of concurrent tasks.
- *Improved Security Model*—Includes a lot of changes to the CLR's security system with the introduction of a new security transparent mode.
- *Networking Improvements*—Include Secure Sockets Layer (SSL) support, Network Address Translation (NAT) traversal support, security improvements for Windows authentication, and support for a null cipher for use in encryption.
- *Improved Core ASP.NET Services*—Includes features, such as Web.config file minification, auto-start Web applications, extensible output caching, multi-targeting, expanding the range of URLs, and so on, that improve the core ASP.NET services.
- *Improvements in WPF 4*—Includes addition of new controls, touch and manipulation technique, improved support for Silverlight, support for multiple platform versions, and visual data binding. In addition to these features, many more enhancements are made in WPF 4 in the areas of graphics, text, binding, and XAML browser applications.
- *Improved Entity Framework (EF)*—Includes foreign keys and lazy loading support, plain old CLR object (POCO) support for entities, and enhanced stored procedure support.
- *Integration between WCF and WF*—Supports stateful and conversational services.

Unmanaged code is the code that is executed directly by the operating system outside the CLR environment. It is directly compiled to native machine code which depends on the machine configuration.

In the managed code, since the execution of the code is governed by CLR, the runtime provides different services, such as garbage collection, type checking, exception handling, and security support. These services help provide uniformity in platform and language-independent behavior of managed code applications. In the unmanaged code, the allocation of memory, type safety, and security is required to be taken care of by the developer. If the unmanaged code is not properly handled, it may result in memory leak. Examples of unmanaged code are ActiveX components and Win32 APIs that execute beyond the scope of native CLR.

#### 12. State the differences between the Dispose () and Finalize () .

- CLR uses the Dispose() and Finalize() methods to perform garbage collection of run-time objects of .NET applications.
- The Finalize() method is called automatically by the runtime. CLR has a garbage collector (GC), which periodically checks for objects in heap that are no longer referenced by any object or program. It calls the Finalize() method to free the memory used by such objects. The Dispose() method is called by the programmer. Dispose is another method to release the memory used by an object. The Dispose() method needs to be explicitly called in code to dereference an object from the heap. The Dispose() method can be invoked only by the classes that implement the IDisposable interface.

#### 13. Name the classes that are introduced in the System.Numerics namespace.

- The following two new classes are introduced in the System.Numerics namespace:
- BigInteger* —Refers to a non-primitive integral type, which is used to hold a value of any size. It has no lower and upper limit, making it possible for you to perform arithmetic calculations with very large numbers, even with the numbers which cannot hold by double or long.
- Complex* —Represents complex numbers and enables different arithmetic operations with complex numbers. A number represented in the form  $a + bi$ , where  $a$  is the real part, and  $b$  is the imaginary part, is a complex number.

#### 14. How do you instantiate a complex number?

- The following are the different ways to assign a value to a complex number:
  - By passing two Double values to its constructor. The first value represents the real, and the second value represents imaginary part of a complex number. For example,

```
Complex C1 = new Complex(5, 8); //C1 represents (5, 8)
```

- By assigning a Byte, SByte, Int16, UInt16, Int32, UInt32, Int64, UInt64, Single, or Double value to a Complex object. The assigned value represents the real part of the complex number, and its imaginary part becomes 0. For example,

```
Complex C2 = 15; //C2 represents (15, 0)
```

- By casting a Decimal or BigInteger value to a Complex object.

For example,

```
Complex C3 = (Complex)10; //C3 represents (10, 0)
```

- Assigning the value returned by an operator to a Complex variable. For example,

```
Complex C4 = 15 + 8i; //C4 represents (15, 8)
```

#### 15. What is lazy initialization?

- Lazy initialization is a process by which an object is not initialized until it is first called in your code. The .NET 4.0 introduces a new wrapper class, `System.Lazy<T>`, for executing the lazy initialization in your application. Lazy initialization helps you to reduce the wastage of resources and memory requirements to improve performance. It also supports thread-safety.

#### 16. Give a brief introduction on side-by-side execution. Can two applications, one using private assembly and the other using the shared assembly be stated as side-by-side executables?

- Side-by-side execution enables you to run multiple versions of an application or component and CLR on the same computer at the same time. As versioning is applicable only to shared assemblies and not to private assemblies, two applications, one using a private assembly and other using a shared assembly, cannot be stated as side-by-side executables.

#### 17. Explain covariance and contra-variance in .NET Framework 4.0. Give an example for each.

- In .NET 4.0, the CLR supports covariance and contravariance of types in generic interfaces and delegates. Covariance enables you to cast a generic type to its base types, that is, you can assign a instance of

type `IEnumerable<T1>` to a variable of type `IEnumerable<T2>` where, `T1` derives from `T2`. For example,

```
IEnumerable<T1> list = new List<string>();
```

Contravariance allows you to assign a variable of `Action <derived>` to a variable of type `Action <base>`. For example,

```
IComparer<object> obj1 = getComparer();
```

```
obj1.Compare(string obj2, obj1);
```

.NET framework 4.0 uses some language keywords (`out` and `in`) to annotate covariance and contra-variance. `Out` is used for covariance, while `in` is used for contra-variance.

Variance can be applied only to reference types, generic interfaces, and generic delegates. These cannot be applied to value types and generic types.

### 18. What is garbage collection? Explain the difference between garbage collections in .NET 4.0 and earlier versions.

- Garbage collection prevents memory leaks during execution of programs. Garbage collector is a low-priority process that manages the allocation and deallocation of memory for your application. It checks for the unreferenced variables and objects. If GC finds any object that is no longer used by the application, it frees up the memory from that object.
- GC has changed a bit with the introduction of .NET 4.0. In .NET 4.0, the `GC.Collect()` method contains the following overloaded methods:

```
GC.Collect(int)
GC.Collect(int, CollectionMode)
```

Another new feature introduced in .NET is to notify you when the `GC.Collect()` method is invoked and completed successfully by using different methods. The .NET 4.0 supports a new background garbage collection that replaces the concurrent garbage collection used in earlier versions. This concurrent GC allocates memory while running and uses current segment (which is 16 MB on a workstation) for that. After that, all threads are suspended. In case of background GC, a separate ephemeral GC - gen0 and gen1 can be started, while the full GC - gen0, 1, and 2 - is already running.

### 19. How many types of generations are there in a garbage collector?

- Memory management in the CLR is divided into three generations that are build up by grouping memory segments. Generations enhance the garbage collection performance. The following are the three types of generations found in a garbage collector:

- Generation 0**—When an object is initialized, it is said to be in generation 0.
- Generation 1**—The objects that are under garbage collection process are considered to be in generation 1.
- Generation 2**—Whenever new objects are created and added to the memory, they are added to generation 0 and the old objects in generation 1 are considered to be in generation2.

### 20. What are tuples?

- Tuple is a fixed-size collection that can have elements of either same or different data types. Similar to arrays, a user must have to specify the size of a tuple at the time of declaration.
- Tuples are allowed to hold up from 1 to 8 elements and if there are more than 8 elements, then the 8th element can be defined as another tuple. Tuples can be specified as parameter or return type of a method.

### 21. How can you instantiate a tuple?

- The following are two ways to instantiate a tuple:
- Using the new operator. For example,

```
Tuple<string, int> = new Tuple<string, int>(“Hello”, 12);
```

- Using the Create factory method available in the Tuple class. For example,

```
Tuple<int, int> = Tuple.Create(2, 4);
```

### 22. What are the improvements made in the file system enumeration in .NET 4.0?

- In .NET 4.0, new enumeration methods are introduced. These methods are included in `Directory` and `DirectoryInfo` classes and return the `IEnumerable<T>` collections instead of arrays. Therefore, you never need to allocate an array before using these enumeration methods.

Similarly, some new methods allow you to read and write lines from files by using the `IEnumerable<T>` interface.

The following examples show how to get the list of files from a directory. It will make you understand the difference between file system enumeration in .NET 4.0 and the earlier versions:

In earlier versions,

```
DirectoryInfo directory = new DirectoryInfo("C:\\");

File[] files = directory.GetFiles();
```

In .NET 4.0,

```
DirectoryInfo directory = new DirectoryInfo
((@"C:\Temp\")
+ Environment.GetFolderPath(Environment.SpecialFolder.Temp)
+ @"\TemporaryFiles");

```

**23. Explain memory-mapped files.**

- Memory-mapped files (MMFs) allow you map the content of a file to the logical address of an application. These files enable the multiple processes running on the same machine to share data with each other. The `MemoryMappedFile.CreateFromFile()` method is used to obtain a `MemoryMappedFile` object that represents a persisted memory-mapped file from a file on disk. These files are included in the `System.IO.MemoryMappedFiles` namespace. This namespace contains four classes and three enumerations to help you access and secure your file mappings.

**24. What is the difference between int and int32.**

- There is no difference between `int` and `int32`. `System.Int32` is a .NET Class and `int` is an alias name for `System.Int32`.

**25. What is code access security (CAS)?**

- Code access security (CAS) is part of the .NET security model that prevents unauthorized access of resources and operations, and restricts the code to perform particular tasks.

**26. How does CAS works?**

- There are two key concepts of CAS security policy- code groups and permissions. A code group contains assemblies in it in a manner that each .NET assembly is related to a particular code group and some permissions are granted to each code group. For example, using the default security policy, a control downloaded from a Web site belongs to the Zone, Internet code group, which adheres to the permissions defined by the named permission set. (Normally, the named permission set represents a very restrictive range of permissions.) Assembly execution involves the following steps:

1. Evidences are gathered about assembly.
2. Depending on the gathered evidences, the assembly is assigned to a code group.
3. Security rights are allocated to the assembly, depending on the code group.
4. Assembly runs as per the rights assigned to it.

**27. How can you turn-on and turn-off CAS?**

- You can use the Code Access Security Tool (`Caspol.exe`) to turn security on and off.

To turn off security, type the following command at the command prompt:  
`caspol -security off`

To turn on security, type the following command at the command prompt:  
`caspol -security on`

In the .NET Framework 4.0, for using Caspol.exe, you first need to set the `<legacyCasPolicy>` element to true.

**28. What are the improvements made in CAS in .NET 4.0?**

- The CAS mechanism in .NET is used to control and configure the ability of managed code. Earlier, as this policy was applicable for only native applications, the security guarantee was limited. Therefore, developers used to look for alternating solutions, such as operating system-level solutions. This problem was solved in .NET Framework 4 by turning off the machine-wide security. The shared and hosted Web applications can now run more securely. The security policy in .NET Framework 4 has been simplified using the transparency model. This model allows you to run the Web applications without concerning about the CAS policies.

As a result of security policy changes in .NET Framework 4.0, you may encounter compilation warnings and runtime exceptions, if you try to use the obsolete CAS policy types and members either implicitly or explicitly. However, you can avoid the warnings and errors by using the `<NetFx40_LegacySecurityPolicy>` configuration element in the runtime settings schema to opt into the obsolete CAS policy behavior.

**29. Which method do you use to enforce garbage collection in .NET?**

- The `System.GC.Collect()` method.

**30. What are code contracts?**

- Code contracts help you to express the code assumptions and statements stating the behavior of your code in a language-neutral way. The contracts are included in the form of pre-conditions, post-conditions and object-invariants. The contracts help you to improve testing by enabling run-time checking, static contract verification, and documentation generation.

The `System.Diagnostics.Contracts` namespace contains static classes that are used to express contracts in your code.

### 3.1. Which is the root namespace for fundamental types in .NET Framework?

- `System.Object` is the root namespace for fundamental types in .NET Framework.

### 3.2. What is managed extensibility framework?

- Managed extensibility framework (MEF) is a new library that is introduced as a part of .NET 4.0 and Silverlight 4. It helps in extending your application by providing greater reuse of applications and components. MEF provides a way for host application to consume external extensions without any configuration requirement.

### 3.3. Is there a way to suppress the finalize process inside the garbage collector forcibly in .NET?

- Use the `GC.SuppressFinalize()` method to suppress the finalize process inside the garbage collector forcibly in .NET.

### 3.4. Mention the execution process for managed code.

- A piece of managed code is executed as follows:
  - Choosing a language compiler
  - Compiling the code to MSIL
  - Compiling MSIL to native code
  - Executing the code

## 2

### Programming Concepts

#### 1. Define variable and constant.

- A variable can be defined as a meaningful name that is given to a data storage location in the computer's memory that contains a value. Every variable associated with a data type determines what type of value can be stored in the variable, for example an integer, such as 10, a decimal, such as 30.05, or a character, such as 'A'.

You can declare variables by using the following syntax:

```
<datatype> <variableName>
```

A constant is similar to a variable except that the value, which you assign to a constant, cannot be changed, as in case of a variable. Constants must be initialized at the same time they are declared. You can declare constants by using the following syntax:

```
const <datatype> <constantName> = <value>;
```

#### 2. What is a data type? How many types of data types are there in .NET.

- A data type is a data storage format that can contain a specific type or range of values. Whenever you declare variables, each variable must be assigned a specific data type. Some common data types include integers, floating point, characters, and strings. The following are the two types of data types available in .NET:

- **Value type** — Refers to the data type that contains the data. In other words, the exact value or the data is directly stored in this data type. It means that when you assign a value type variable to another variable, then it copies the value rather than copying the reference of that variable. When you create a value type variable, a single space in memory is allocated to store the value. Primitive data types, such as int, float, and char are examples of value type variables.
- **Reference type** — Refers to a data type that can access data by reference. Reference is a value or an address that accesses a particular data by address, which is stored elsewhere in memory. You can say that reference is the physical address of data.

where the data is stored in memory or in the storage device. Some built-in reference types variables in .Net are string, array, and object.

### 3. Mention the two major categories that distinctly classify the variables of C# programs.

- Variables that are defined in a C# program belong to two major categories: **value type** and **reference type**. The variables that are based on value type contain a value that is either allocated on a stack or allocated in-line in a structure. The variables that are based on reference types store the memory address of a variable, which in turn stores the value and are allocated on the heap. The variables that are based on value types have their own copy of data and therefore operations done on one variable do not affect other variables. The reference-type variables reflect the changes made in the referring variables.

Predict the output of the following code segment:

```
int x = 72;
int y = 12;
int w;
object o;
o = x;
w = y;
Console.WriteLine(w);
```

The output of the code is **504**.

### 4. Which statement is used to replace multiple if-else statements in code.

- In Visual Basic, the **Select—Case** statement is used to replace multiple **If – Else** statements and in C#, the **switch-case** statement is used to replace multiple **if-else** statements.

### 5. What is the syntax to declare a namespace in .NET?

- In .NET, the **namespace** keyword is used to declare a namespace in the code.

The syntax for declaring a namespace in C# is:

```
namespace UserNamespace;
```

The syntax for declaring a namespace in VB is:

```
NameSpace UserNamespace;
```

### 6. Determine the output of following code snippet:

```
int a = 29;
a--;
a = -a;
Console.WriteLine ("The value of a is: {0}", a);
```

The output of the code is **-1**.

### 7. Briefly explain the characteristics of value-type variables that are supported in the C# programming language.

- The variables that are based on value types directly contain values.
- The characteristics of value-type variables that are supported in C# programming language are as follows:
  - All value-type variables derive implicitly from the System.ValueType class
  - You cannot derive any new type from a value type
  - Value types have an implicit default constructor that initializes the default value of that type
  - The value type consists of two main categories:
    - Structs**—Summarizes small groups of related variables
    - Enumerations**—Consists of a set of named constants

### 8. Briefly explain the characteristics of reference-type variables that are supported in the C# programming language.

- The variables that are based on reference types store references to the actual data. The keywords that are used to declare reference types are:
  - Class**—Refers to the primary building block for the programs, which is used to encapsulate variables and methods into a single unit
  - Interface**—Contains only the signatures of methods, properties, events, or indexers
  - Delegate**—Refers to a reference type that is used to encapsulate a named or anonymous method

### 9. Differentiate between Boxing and Unboxing.

- When a value type is converted to an object type, the process is known as boxing; whereas, when an object type is converted to a value type, the process is known as unboxing. The following code-snippet shows the use of the object type:

```
object x;
x=15; //Boxing
```

In the preceding code-snippet, the first line of the code creates a variable x of the object type and the second line assigns a value of 15 to the variable x. This process of assigning the converted value 15 to the object data type x is called boxing.

### 10. Mention different types of comments that can be inserted in a code.

- Comments are brief but explanatory notes added to the code for the benefit of those reading it. You can add comments to the code to

make it more understandable; however, a commented code does not execute. The C# compiler considers all the text on a line following two slashes (//) to be comment. It skips every text written after // in the line.

In C#, you can comment a code in two ways:

- **Single-line comment** —Refers to one-line comments that are given in a code. For example, consider the following code snippet:  

```
Console.WriteLine("Hello"); //this is a welcome message
```

The output of the above text will be Hello. The comment just highlights the purpose of the line of code.
- **Multi-line comment** —Refers to comments that contain multiple lines. You can use multi-line comments when you want the compiler to ignore multiple lines in a single instance. For example, consider the following code snippet:  

```
/* This is an example of using comments
 * in C# code. You can use comments
 * for better understandability */

```

In a multi-line comment, the compiler ignores the text written between the symbols /\* and \*/.

You can also comment on your Visual Basic code using an apostrophe (') mark. Visual Basic ignores whatever follows the apostrophe on the line. For example, consider the following code snippet:

```
Console.WriteLine("Hello"); 'this is a welcome message
```

The output of the preceding code snippet will be Hello. The comment just highlights the purpose of the line of code.

### 11. What is an identifier?

- Identifiers are nothing but names given to various entities uniquely identified in a program. The name of identifiers must differ in spelling or casing. For example, MyIndia and myIndia are two different identifiers. Programming languages, such as C# and Visual Basic, strictly restrict the programmers from using any keyword as identifiers. Programmers cannot develop a class whose name is public, because, public is a keyword used to specify the accessibility of data in programs.

Some examples of valid identifiers are as follows:

```
Employee
CalculatePay
```

Some examples of invalid identifiers are given as follows:

```
123TheAnswer //starts with a digit
@002456 //contains a non-alphanumeric character
In Code //contains a keyword
continue //this is a keyword
```

### 12. What is the difference between constants and read-only variables that are used in programs?

- Constants perform the same tasks as read-only variables with some differences. The differences between constants and read-only variables are listed in Table 2.1.

Table 2.1: Differentiating between constants and read-only variables

Constants	Read-only variables
-----------	---------------------

Constants support value-type variables	Read-only variables can hold reference-type variables
Constants are dealt with at compile-time.	Read-only variables are evaluated at runtime.

### 13. Explain keywords with example.

- Keywords are those words that are reserved to be used for a specific task. These words cannot be used as identifiers. You cannot use a keyword to define the name of a variable or method. Keywords are used in programs to use the features of object-oriented programming. For example, the abstract keyword is used to implement abstraction and the inherits keyword is used to implement inheritance by deriving subclasses in C# and Visual Basic, respectively. The new keyword is universally used in C# and Visual Basic to implement encapsulation by creating objects.

### 14. What are the different types of literals?

- A literal is a textual representation of a particular value of a type. The different types of literals in Visual Basic are:
  - **Boolean Literals** —Refers to the True and False literals that map to the true and false state, respectively.
  - **Integer Literals** —Refers to literals that can be decimal (base 10), hexadecimal (base 16), or octal (base 8).
  - **Floating-Point Literals** —Refers to an integer literal followed by an optional decimal point. By default, a floating-point literal is of type Double.

- String Literals**—Refers to a sequence of zero or more Unicode characters beginning and ending with an ASCII double-quote character.
- Character Literals**—Represents a single Unicode character of the Char type.
- Date Literals**—Represents time expressed as a value of the Date type.
- Nothing**—Refers to a literal that does not have a type and is convertible to all types in the type system.
- The different types of literals in C# are:
- Boolean Literals**—Refers to the True and False literals that map to the true and false states, respectively.
  - Integer Literals**—Refers to literals that are used to write values of types int, uint, long, and ulong.
  - Real Literals**—Refers to literals that are used to write values of types float, double, and decimal.
  - Character Literals**—Represents a single character that usually consists of a character in quotes, such as 'a'.
  - String Literals**—Refers to string literals, which can be of two types in C#:
- A regular string literal consists of zero or more characters enclosed in double quotes, such as "hello".
  - A verbatim string literal consists of the @ character followed by a double-quote character, such as @("hello").
  - The Null Literal**—Represents the null-type.

#### 15. Give the syntax of using the for loop in C# code?

- The syntax of using the for loop in C# code is given as follows:

```
for (_initializer; condition; loop expression)
{
    // statements
}
```

In the preceding syntax, initializer is the initial value of the variable, condition is the expression that is checked before the execution of the for loop, and loop expression either increments or decrements the loop counter.

The example of using the for loop in C# is shown in the following code snippet:

```
for (int i = 0; i < 5; ++i)
    Console.WriteLine("Hello");
```

In the preceding code snippet, the word Hello will be displayed for five times in the output window.

#### 16. Give the syntax of using the while loop in a C# program.

- You can use the while loop to execute a statement or a block of statements until the specified condition evaluates to false. The best situation to use the while loop in your program is when you do not know in advance how many times the loop needs to be executed. The condition in the while loop is evaluated before executing the loop. The loop continues to execute as long as the condition remains true. This implies that if the condition in the while loop evaluates to true, then the statements inside the while loop are executed; otherwise, not. Once the statements inside the while loop are executed, the program control is transferred to the starting point of the while loop to check the condition again for the next iteration.
- The syntax of using the while loop in C# is:

```
while (condition)
{
    // statements
}
```

You can find an example of using the while loop in C#:

```
int i = 0;
while (i < 5)
{
    Console.WriteLine(i);
    i++;
}
```

The output of the preceding code is: 0, 1, 2, 3, and 4.

#### 17. Differentiate between the while and for loop in C#.

- The while and for loops are used to execute those units of code that need to be repeatedly executed, unless the result of the specified condition evaluates to false. The only difference between the two is in their syntax. The for loop is distinguished by setting an explicit loop variable.

#### 18. What are the different iteration statements in Visual Basic?

- An iteration statement executes a statement or a set of statements in a repeated manner. In Visual Basic, there are four types of iteration statements, which are as follows:
- While Statement**—Executes a set of statements as long as a given condition evaluates to true. The syntax of using the While statement in Visual Basic is:

```
While condition
{
    statements
}
End While
```

In the preceding syntax, the statements enclosed in the While statement are repeatedly executed till the condition evaluates to true. You can also terminate a While statement at any time with an Exit While statement.

- **The Do While statement**—Executes different set of statements for variable number of times. The syntax of using the Do While statement in Visual Basic is as follows:

**Do** [ **While** [ **untill** ] **condition**

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Until** ] **condition**

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

[ **Statements** ]

**Loop**

-or-

**Do**

[ **Statements** ]

**Loop** [ **Do** ]

Access Modifiers	Description	Example
protected internal	Gives access to the members that are visible either to the current assembly or to the types derived from the class in which they are declared.	

Different types of access modifiers in Visual Basic are listed in Table 2.3.

Table 2.3: Access Modifiers in Visual Basic

Access Modifiers	Description	Syntax
Public	Gives unrestricted access to members from both inside and outside a class.	Public Class A
Protected	Gives protected access to members. You can access protected members from either the class in which they are declared or a class derived from the class in which they are declared.	Protected Class A
Friend	Gives access to the members that are in the same assembly. If a member with the Friend access modifier is accessed from outside the assembly in which it has been defined, an error is generated.	Friend A As String
Protected Friend	Gives access to members that are visible to all the derived classes within the same assembly.	Protected Friend A As String
Private	Gives access to the members that are within the body of the class in which they are declared.	Private A As Integer

We can also provide arguments in a sub-procedure; however, it does not return a new value.

The function is also a set of statements within the Function and End Function statements. It is similar to sub-procedure and performs the same task. The main difference between a function and a sub-procedure is that sub-procedures do not return a value while functions do.

## 22. What does a break statement do in the switch statement?

- The switch statement is a selection control statement that is used to handle multiple choices and transfer control to the case statements within its body. The following code snippet shows an example of the use of the switch statement in C#:

```
switch (choice)
{
    case 1:
        Console.WriteLine("First");
        break;
    case 2:
        Console.WriteLine("Second");
        break;
    default:
        Console.WriteLine("Wrong choice");
        break;
}
```

In switch statements, the break statement is used at the end of a case statement. The break statement is mandatory in C# and it avoids the fall through of one case statement to another.

## 23. Write a program to swap two numbers and print them in C#.

- The program to swap two numbers and print them is as follows:

```
using System;
class Swap_Three_No
{
    static void Main()
    {
        int a, b;
        int temp = 0;
        void accept()
        {
            Console.WriteLine("Enter two number");
            Console.WriteLine("Enter first");
            Console.WriteLine("Before swap First");
            Console.WriteLine("Enter second");
            Console.WriteLine("Before swap Second");
            Console.WriteLine("Enter third");
            Console.WriteLine("Before swap Third");
        }
        accept();
        a = temp;
        b = temp;
        temp = a;
        a = b;
        b = temp;
        Console.WriteLine("After swap First");
        Console.WriteLine("After swap Second");
        Console.WriteLine("After swap Third");
    }
}
```

## 21. What is the main difference between sub-procedure and function?

- The sub-procedure is a block of multiple visual basic statements within Sub and End Sub statements. It is used to perform certain tasks, such as changing properties of objects, receiving or processing data, and displaying an output. You can define a sub-procedure anywhere in a program, such as in modules, structures, and classes.

and  
b-temp

SwapThreeNo st = new SwapThreeNo();  
st.accept();  
st.display();

3

## Object-Oriented Programming

### 1. What is object-oriented programming (OOP)?

- OOP is a technique to develop logical modules, such as classes that contain properties, methods, fields, and events. An object is created in the program to represent a class. Therefore, an object encapsulates all the features, such as data and behavior that are associated to a class. OOP allows developers to develop modular programs and assemble them as software. Objects are used to access data and behaviors of different software modules, such as classes, namespaces, and sharable assemblies. .NET Framework supports only OOP languages, such as Visual Basic .NET, Visual C#, and Visual C++.

### 2. What is a class?

- A class can be defined as the primary building block of OOP. It also serves as a template that describes the properties, state, and behaviors common to a particular group of objects.
- A class contains data and behavior of an entity. For example, the aircraft class can contain data, such as model number, category, and color and behavior, such as duration of flight, speed, and number of passengers. A class inherits the data members and behaviors of other classes by extending from them.

The example of optional parameter is as follows:

```
public static void Add(int a, int b, int c = 0, int d = 0);
{
    // code
}
```

```
Add(10, 20); // 10 + 20 + 0 + 0
Add(10, 20, 30); // 10 + 20 + 30 + 0
Add(10, 20, 30, 40); // 10 + 20 + 30 + 40
```

*Named parameter* —Refers to the new parameter introduced in C# 4.0. Now you can provide arguments by name rather than position.

The example of the named parameter is as follows:

```
public void CreateAccount(string name, string
address, int unknown, int age = 0)
{
    CreateAccount("Kunal", "age", 28);
    CreateAccount(address, "India", name, "Kunal");
}
```

- A class acts as a blue-print that defines the properties, states, and behaviors that are common to a number of objects. An object is an instance of the class. For example, you have a class called Vehicle and Car is the object of that class. You can create any number of objects for the class named Vehicle, such as Van, Truck, and Auto.

- The new operator is used to create an object of a class. When an object of a class is instantiated, the system allocates memory for every data member that is present in the class.

#### 5. Explain the basic features of OOPS.

- The following are the four basic features of OOP:
  - Polymorphism* —Allows you to use an entity in multiple forms.
  - Encapsulation* —Prevents the data from unwanted access by binding of code and data in a single unit called object.
  - Inheritance* —Promotes the reusability of code and eliminates the use of redundant code. It is the property through which a child class obtains all the features defined in its parent class.
  - When a class inherits the common properties of another class, the class inheriting the properties is called a derived class and the class that allows inheritance of its common properties is called a base class.

#### 6. What are abstract classes? What are the distinct characteristics of an abstract class?

- An abstract class is a class that cannot be instantiated and is always used as a base class.
  - The following are the characteristics of an abstract class:
    - You cannot instantiate an abstract class directly. This implies that you cannot create an object of the abstract class; it must be inherited.
    - You can have abstract as well as non-abstract members in an abstract class.
    - You must declare at least one abstract method in the abstract class.
    - An abstract class is always public.
    - An abstract class is declared using the **abstract** keyword.

The basic purpose of an abstract class is to provide a common definition of the base class that multiple derived classes can share.

#### 7. When do you really need to create an abstract class?

- We define abstract classes when we define a template that needs to be followed by all the derived classes.

#### 8. State the features of an interface.

- An interface is a template that contains only the signature of methods. The signature of a method consists of the numbers of parameters, the type of parameter (value, reference, or output), and the order of parameters. An interface has no implementation on its own because it contains only the definition of methods without any method body. An interface is defined using the **Interface** keyword. Moreover, you cannot instantiate an interface. The various features of an interface are as follows:
  - An interface is used to implement multiple inheritance in code. This feature of an interface is quite different from that of abstract classes because a class cannot derive the features of more than one class but can easily implement multiple interfaces.
  - It defines a specific set of methods and their arguments.
  - Variables in interface must be declared as public, static, and final while methods must be public and abstract.
  - A class implementing an interface must implement all of its methods.
  - An interface can derive from more than one interface.

#### 9. Differentiate between an abstract class and an interface.

- The differences between an abstract class and an interface are listed in Table 3.1.

Table 3.1: Differences between an Abstract Class and an Interface

Abstract Class	Interface
<input type="checkbox"/> A class can extend only one abstract class	<input checked="" type="checkbox"/> All interface can only have public interfaces
<input checked="" type="checkbox"/> The members of an abstract class can be private as well as protected	<input type="checkbox"/> Abstract classes should have subclasses
<input checked="" type="checkbox"/> Methods in an abstract class can only have public members	<input type="checkbox"/> Only an interface can extend another interface
<input type="checkbox"/> Interfaces must have implementations by classes	<input type="checkbox"/> There can be a construction for abstract class

Abstract Class	Interface
The class extending the abstract class may or may not implement any of its method	All methods of interface need to be implemented by a class implementing that interface

An abstract class can implement methods	Interfaces cannot contain body of methods
---	---

#### 10. What are methods?

- Methods are the building blocks of a class, in which they are linked together to share and process data to produce the result. In other words, a method is a block of code that contains a series of statements and represents the behavior of a class. While declaring a method you need to specify the access specifier, the return value, the name of the method, and the method parameters. All these combined together is called the signature of the method.

#### 11. Give a brief description of properties in C# and the advantages that are obtained by using them in programs.

- In C#, a property is a way to expose an internal data element of a class in a simple and intuitive manner. In other words, it is a simple extension of data fields. You can create a property by defining an externally available name and then writing the set and get property accessors. The get property accessor is used to return the property value. The set property accessor is used to assign a new value to the property.

#### 12. What do you mean by data encapsulation?

- Data encapsulation is a concept of binding data and code in single unit called object and hiding all the implementation details of a class from the user. It prevents unauthorized access of data and restricts the user to use the necessary data only.

#### 13. Explain different types of inheritance.

- Inheritance in OOP is of four types:
  - *Single inheritance* —Contains one base class and one derived class
  - *Hierarchical inheritance* —Contains one base class and multiple derived classes of the same base class
  - *Multilevel inheritance* —Contains a class derived from a derived class
  - *Multiple inheritance* —Contains several base classes and a derived class

#### 14. Explain the concept of constructor?

- Constructor is a special method of a class, which is called automatically when the instance of a class is created. It is created with the same name as the class and initializes all class members, whenever you access the class. The main features of a constructor are as follows:
  - Constructors do not have any return type
  - Constructors are always public
  - It is not mandatory to declare a constructor; it is invoked automatically by .NET Framework

#### 15. Explain the concept of destructor?

- A destructor is a special method for a class and is invoked automatically when an object is finally destroyed. The name of the destructor is also same as that of the class but is followed by a prefix tilde (~).
    - A destructor is used to free the dynamic allocated memory and release the resources. You can, however, implement a custom method that allows you to control object destruction by calling the destructor.
- The main features of a destructor are as follows:
- Destructors do not have any return type
  - Similar to constructors, destructors are also always public
  - Destructors cannot be overloaded

#### 16. What is a static constructor?

- Static constructors are introduced with C# to initialize the static data of a class. CLR calls the static constructor before the first instance is created.
 

The static constructor has the following features:

  - No access specifier is required to define it
  - You cannot pass parameters in static constructor
  - A class can have only one static constructor
  - It can access only static members of the class
  - It is invoked only once, when the program execution begins

All .NET languages supports single, hierarchical, and multilevel inheritance. They do not support multiple inheritance because in these languages, a derived class cannot have more than one base class. However, you can implement multiple inheritance in .NET through interfaces.

**17. You have defined a destructor in a class that you have developed by using the C# programming language, but the destructor never executed. Why did the destructor not execute?**

- The runtime environment automatically invokes the destructor of a class to release the resources that are occupied by variables and methods of an object. However, in C#, programmers cannot control the timing for invoking destructors, as Garbage Collector is only responsible for releasing the resources used by an object. Garbage Collector automatically gets information about unreferenced objects from .NET's runtime environment and then invokes the Finalize() method.

Although, it is not preferable to force Garbage Collector to perform garbage collection and retrieve all inaccessible memory, programmers can use the Collect() method of the Garbage Collector class to forcefully execute Garbage Collector.

**18. Why is the virtual keyword used in code?**

- The virtual keyword is used while defining a class to specify that the methods and the properties of that class can be overridden in derived classes.

**19. Does .NET support multiple inheritance?**

- .NET does not support multiple inheritance directly because in .NET, a class cannot inherit from more than one class. .NET supports multiple inheritance through interfaces.

**20. How can you prevent your class to be inherited further?**

- You can prevent a class from being inherited further by defining it with the sealed keyword.

**21. A structure in C# can implement one or more interfaces. Is it true or false?**

- Yes, it is true. Like classes, in C#, structures can implement one or more interfaces.

**22. What is a delegate?**

- A delegate is similar to a class that is used for storing the reference to a method and invoking that method at runtime, as required. A delegate can hold the reference of only those methods whose signatures are same as that of the delegate. Some of the examples of delegates are type-safe functions, pointers, or callbacks.

**23. Define an event.**

- Whenever an action takes place in a class, that class provides a notification to other classes or objects that are assigned to perform particular tasks. These notifications are called events. For example, when a button is clicked, the class generates an event called Click. An event can be declared with the help of the **event** keyword.

**24. What is a multicast delegate?**

- Each delegate object holds reference to a single method. However, it is possible for a delegate object to hold references of and invoke multiple methods. Such delegate objects are called multicast delegates or combinable delegates.

**25. Do events have return type?**

- No, events do not have return type.

**26. How can you prevent a class from overriding in C# and Visual Basic?**

- You can prevent a class from overriding in C# by using the **sealed** keyword; whereas, the **NotInheritable** keyword is used to prevent a class from overriding in Visual Basic.

**27. What is the syntax to inherit from a class in C#?**

- When a class is derived from another class, then the members of the base class become the members of the derived class. The access modifier used while accessing members of the base class specifies the access status of the base class members inside the derived class. The syntax to inherit a class from another class in C# is as follows:

**28. Can you allow a class to be inherited, but prevent a method from being overridden in C#?**

- Yes. Just declare the class **public** and make the method **sealed**.

**29. How is method overriding different from method overloading?**

- Overriding involves the creation of two or more methods with the same name and same signature in different classes (one of them should be parent class and other should be child). Overloading is a concept of using a method at different places with same name and different signatures within the same class.

**30. Can you declare an overridden method to be static if the original method is not static?**

- No. Two virtual methods must have the same signature.

### 31. What are the different ways a method can be overloaded?

- The different ways to overload a method are given as follows:
  - By changing the number of parameters used
  - By changing the order of parameters
  - By using different data types for the parameters

### 32. Is it possible for a class to inherit the constructor of its base class?

- No, a class cannot inherit the constructor of its base class.

### 33. What is the difference between procedural and object-oriented programming?

- Procedural programming is based upon the modular approach in which the larger programs are broken into procedures. Each procedure is a set of instructions that are executed one after another. On the other hand, OOP is based upon objects. An object consists of various elements, such as methods and variables. Access modifiers are not used in procedural programming, which implies that the entire data can be accessed freely anywhere in the program. In OOP, you can specify the scope of a particular data by using access modifiers—public, private, internal, protected, and protected internal.

### 34. Can you specify the accessibility modifier for methods inside the interface?

- All the methods inside an interface are always **public**, by default. You cannot specify any other access modifier for them.

### 35. Can you declare a private class in a namespace?

- The classes in a namespace are **internal**, by default. However, you can explicitly declare them as **public** only and not as **private**, **protected**, or **protected internal**. The nested classes can be declared as **private**, **protected**, or **protected internal**.

### 36. Can you inherit private members of a class?

- No, you cannot inherit private members of a class because private members are accessible only to that class and not outside that class.

### 37. What is a namespace?

- Namespace is considered as a container that contains functionally related group of classes and other types.

### 38. What is the difference between a class and a structure?

- The differences between a class and a structure are listed in Table 3.2.

Table 3.2: Differences between a class and a structure

Class	Structure
The members of a class are <b>reference</b> type.	The members of a structure are <b>value</b> type.
While instantiating a class, CLR allocates memory for its instance in heap.	In structure, memory is allocated on stack.
Classes support inheritance.	Structures do not support inheritance.
Variables of a class can be assigned as null.	Structure members cannot have null values.
Class can contain constructor/destructor.	Structure does not require constructor/destructor and members can be initialized automatically.

### 39. What are similarities between a class and a structure?

- Structures and classes are the two most important data structures that are used by programmers to build modular programs by using OOP languages, such as Visual Basic .NET, and Visual C#. The following are some of the similarities between a class and a structure:
  - Access specifiers, such as **public**, **private**, and **protected**, are identically used in structures and classes to restrict the access of their data and methods outside their body.
  - Both can have constructors, methods, properties, fields, constants, enumerations, events, and event handlers.
  - Both structures and classes can implement interfaces to use multiple-inheritance in code.
  - Both structures and classes can have constructors with parameter.
  - Both structures and classes can have delegates and events.

### 40. Define an array.

- An array is defined as a homogeneous collection of elements, stored at contiguous memory locations, which can be referred by the same variable name. All the elements of an array variable can be accessed

by index values. An Index value specifies the position of a particular element in an array variable.

#### 41. What is the index value of the first element in an array?

- In an array, the index value of the first element is 0.

**42. In which namespace, all .NET collection classes are contained?**

- The **System.Collections** namespace contains all the collection classes.

#### 43. What are collections and generics?

- A collection can be defined as a group of related items that can be referred to as a single unit. The **System.Collections** namespace provides you with many classes and interfaces. Some of them are: ArrayList, List, Stack, ICollection, IEnumerable, and IDictionary. Generics provide the type-safety to your class at the compile time. While creating a data structure, you never need to specify the data type at the time of declaration. The **System.Collections.Generic** namespace contains all the generic collections.

#### 44. What are queues and stacks?

- Stacks** refer to a list in which all items are accessed and processed on the Last-In-First-Out (LIFO) basis. In a stack, elements are inserted (push operation) and deleted (pop operation) from the same end called **top**. **Queues** refer to a list in which insertion and deletion of an item is done on the First-In-First-Out (FIFO) basis. The items in a queue are inserted from the one end, called the **rear** end, and are deleted from the other end, called the **front** end of the queue.

#### 45. What are structures?

- Structure is a heterogeneous collection of elements referenced by the same name. A structure is declared using the **struct** keyword. The following is an example that creates a structure to store an employee's information:

```
struct emp
{
    Fixed int empId[15];
    Fixed char name[30];
    Fixed char addr[50];
    Fixed char dept[15];
    Fixed char desg[15];
}
```

The preceding example defines a structure **emp** and the members of this structure specify the information of an employee.

#### 46. What is the difference between arrays and collection?

- The differences between an array and a collection are listed in Table 3.3.

Table 3.3: Differences between an array and a collection

Array	Collection
You need to specify the size of an array at the time of its declaration. It cannot be resized dynamically.	The size of a collection can be adjusted dynamically, as per the user's requirement. It does not have fixed size.
The members of an array should be of the same datatype.	Collection can have elements of different data types.

#### 47. Define enumeration?

- Enumeration is defined as a value type that consists of a set of named values. These values are constants and are called enumerators. An enumeration type is declared using the **enum** keyword. Each enumerator in an enumeration is associated with an underlying type that is set, by default, on the enumerator. The following is an example that creates an enumeration to store different varieties of fruits:

```
enum Fruits {Mango, Apple, Orange, Guava};
```

In the preceding example, an enumeration **Fruits** is created, where number 0 is associated with Mango, number 1 with Apple, number 2 with Orange, and number 3 with Guava. You can access the enumerators of an enumeration by these values.

#### 48. What is a hashtable?

- Hashtable is a data structure that implements the **IDictionary** interface. It is used to store multiple items and each of these items is associated with a unique string key. Each item can be accessed using the key associated with it. In short, hashtable is an object holding the key-value pairs.

#### 49. What is the function of the Try-Catch-Finally block?

- The **try** block encloses those statements that can cause exception and the **catch** block handles the exception, if it occurs. Catch block contains the statements that have to be executed, when an exception occurs. The **finally** block always executes, irrespective of the fact whether or not an exception has occurred. The **finally** block is generally used to perform the cleanup process. If any exception occurs in the **try** block, the program control directly transfers to its corresponding **catch** block and later to the **finally** block. If no

exception occurs inside the **try** block, then the program control transfers directly to the **finally** block.

#### 50. Is it a good practice to handle exceptions in code?

- Yes, you must handle exceptions in code so that you can deal with any unexpected situations that occur when a program is running. For example, dividing a number by zero or passing a string value to a variable that holds an integer value would result in an exception.

#### 51. Is it possible to execute two catch blocks?

- You are allowed to include more than one catch block in your program; however, it is not possible to execute them in one go. Whenever, an exception occurs in your program, the correct catch block is executed and the control goes to the finally block.

#### 52. Can users define their own exceptions in code?

- Yes, customized exceptions can be defined in code by deriving from the System.Exception class.

#### 53. Can you use the throws clause to raise an exception?

- No, the throws clause cannot be used to raise an exception.

The throw statement signals the occurrence of an exception during the execution of a program. When the program encounters a throw statement, the method terminates and returns the error to the calling method.

#### 54. How has exception handling changed in .NET Framework 4.0?

- In .NET 4.0, a new namespace, System.Runtime.ExceptionsServices, has been introduced which contains the following classes for handling exceptions in a better and advanced manner:
  - **HandleProcessCorruptedStateExceptionsAttribute Class** —Enables managed code to handle the corrupted state exceptions that occur in an operating system. These exceptions cannot be caught by specifying the try...catch block. To handle such exceptions, you can apply this attribute to the method that is assigned to handle these exceptions.
  - **FirstChanceExceptionEventArgs Class** —Generates an event whenever a managed exception first occurs in your code, before the common language runtime begins searching for event handlers.

# 4

## Windows Control

### 1. How can we auto size a button to fit its text?

- The Button control has the AutoSize property, which can be set to true or false. If we set the value of the AutoSize property to true, then the button control automatically alters its size according to the content displayed on it.

### 2. Explain the method adopted to customize the visual appearance of the Button control in C#.

- The Button class defines the functionalities of the Button control in .NET Framework. We can create a new class that extends from the Button class, as shown in the following code snippet:

```
public class MyButton : System.Windows.Forms.Button
{
    public MyButton()
    {
        this.Paint += new
System.Windows.Forms.PaintEventHandler
(this.button1_Paint);
    }
    private void button1_Paint(object
sender, System.Windows.Forms.PaintEventArgs e)
{
    //custom drawing
    Pen pen2 = new Pen(Color.Red);
    pen2.Width = 8;
    e.Graphics.DrawLine(pen2, 7, 4, 7,
this.Height - 40);
    pen2.Width = 1;
    e.Graphics.DrawElipse(pen2, this.Width
- 16, 6, 8, 8);
}
}
```

In the preceding code snippet, we have declared an object of the PaintEventHandler class and added a custom Paint event handler method to perform custom drawing.

**3. How can we display an icon or a bitmap image on the Button control?**

- The Button class contains the Image property, which is used to set an image on the Button control. We can also set the alignment of the image by using the ImageAlign property of the Button class.

**4. Which method is used to generate the click event of the Control class for the Button control in C#?**

- The PerformClick() method of the Button class is used to generate the Click event of the System.Windows.Forms.Control class.

**5. A Windows Form will not show the Minimize, Maximize, and Close buttons, if the ControlBox property of the form is set to False. (True/False)**

- True.

**6. How is anchoring different from docking?**

- Docking refers to attaching a control to either an edge (top, right, bottom, or left) or the client area of the parent control. On the other hand, anchoring is a process in which you need to specify the distance that each edge of your control maintains from the edges of the parent control.

**7. How can we disable the context menu for a TextBox control?**

- The TextBox class contains the ContextMenuStrip property. When we set this property to a dummy instance of the ContextMenu class, the TextBox control is unable to provide any context menu on the right-click of the mouse.

**8. How do we format numbers, dates, and currencies in a text box?**

- Each type has a ToString method that can be used to format date, currencies, and numbers. You can also use the String.Format method to format these things as well. To format dates, use the ToString member of the DateTime type.

**9. Write the code that enables to browse for a text file in a system and get its content in a text box control in C#.**

- .NET Framework provides the OpenFileDialog class, which is used to open the open file dialog box. By using the open file dialog box, you can browse local drives to search for a desired file. To perform any file operations in code, such as, opening a file and reading its

**Content, you need to include the System.IO namespace and use a StreamReader class, as shown in the following code snippet:**

```
using System.Text;
using System.IO;
private void button1_Click(object sender,
    System.EventArgs e)
{
    OpenFileDialog dg = new
        OpenFileDialog();
    dg.Filter = "txt files (*.txt)|*.txt|All
    files (*.*)|*.*";
    if(dg.ShowDialog() == DialogResult.OK)
    {
        StreamReader sr =
            File.OpenText(dg.FileName);
        string s = sr.ReadLine();
        StringBuilder sb = new
            StringBuilder();
        while (s != null)
        {
            sb.Append(s);
            s = sr.ReadLine();
        }
        sr.Close();
        textBox1.Text = sb.ToString();
    }
}
```

**10. How can you enforce a TextBox to display characters in uppercase?**

- The TextBox class contains the CharacterCasing property, which is used to specify the case of the content for a Text box. This property accepts a value from the CharacterCasing enumeration of .NET Framework. The members specified in the CharacterCasing enumeration are Lower, Upper, and Normal. You can select any one of these enumerations as a value for the CharacterCasing property of a specified TextBox, as shown in the following code snippet:

```
textBox1.CharacterCasing = CharacterCasing.Upper;
```

**11. How can you enable a TextBox to change its characters format, so that users can enter password?**

- You can set the PasswordChar property of the TextBox class to True to enable it to accept passwords. The code to change the PasswordChar property of the TextBox class is given as follows:

```
textBox1.PasswordChar = '#';
```

- 12. How can you enable a read-only text box to ignore mouse down events so that users can scroll the text and position the cursor in .NET 4.0?**

- You can enable this by extending the TextBox class and overriding the WndProc() method, as shown in the following code snippet:

*Code for VB:*

```
Public Class MyTextBox
    Inherits TextBox
    Protected Overrides Sub WndProc(ByRef m As System.Windows.Forms.Message)
        If (m.Msg = &H&1 OrElse m.Msg = &H&101) Then
            Return ' ignore it
        End If
        MyBase.WndProc(m)
    End Sub
End Class 'MyTextBox
```

*Code for C#:*

```
public class MyTextBox : TextBox
{
    protected override void WndProc(ref System.Windows.Forms.Message m)
    {
        // This call is required by the Windows Forms
        // Form Designer. It must remain at the top of the
        // InitializeComponent() call.
        if ((m.Msg == WM_NCLBUTTONDOWN_WM_LBUTTONDOWN
            || m.Msg == 0x201) && (this.ReadOnly && (m.Msg == 0xA1 | m.Msg == 0x201)))
        {
            return; // ignore it
        }
        base.WndProc(ref m);
    }
}
```

- 13. Which event of a TextBox control helps in restricting a Textbox from accepting numeric digits in .NET 4.0?**

- The KeyPress event of a text box is used to restrict it from accepting numeric digits or any other character.

- 14. When you set a TextBox to ReadOnly, the text color is gray. How can you change this color to the one specified in the ForeColor property of the Textbox in .NET 4.0?**

- You can override the OnPaint event of the text box, as shown in the following code snippet:

```
protected override void OnPaint(PaintEventArgs e)
{
    SolidBrush drawBrush = new
    SolidBrush(ForeColor); //use the
    drawBrush.DrawString(textBox1.Text, new
    Font("Times New Roman", 12), drawBrush, e.ClipRectangle);
```

*Code for C#:*

```
private void Form1_Load(object sender, EventArgs e)
{
    textBox1.AcceptsReturn = false;
}
private void textBox1_KeyDown(object sender,
    KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        EndIf();
    }
    EndSub();
}
```

*Code for VB:*

```
Private sub button1_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles Button1.Click
    TextBox1.AcceptsReturn = False
End Sub
Private sub textBox1_KeyDown(ByVal sender As
    System.Object, ByVal e As System.Windows.Forms.KeyEventArgs)
    Handles
    textBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        EndIf()
    EndSub()
End If
End Sub
```

*Code for C#:*

```
private void Form1_Load(object sender, EventArgs e)
{
    textBox1.AcceptsReturn = false;
}
private void textBox1_KeyDown(object sender,
    KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        EndIf();
    }
    EndSub();
}
```

**16. Write the code to create and save a Rich Text Format (RTF) file in C#.**

■ In C#, you can save an RTF file by using the RichTextBox.SaveFile method, as shown in the following code snippet:

```

private void button1_Click(object sender,
    System.EventArgs e)
{
    // Create a saveFileDialog & initialize the RTF
    extension
    SaveFileDialog saveFileDialog1 = new
    SaveFileDialog();
    saveFileDialog1.DefaultExt = "*.*";
    saveFileDialog1.Filter = "RTF Files (*.rtf)|*.*";
    if(saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // Save the RTF contents of the
        // RichTextBox control that
        // was dragged onto the window Form and
        // published
        richTextBox1.SaveFile(saveFileDialog1.FileName,
            RichTextBoxStreamType.RichText);
        // use to save
        // RTF tags in file
    }
}

```

**17. How can you get the text of the RichTextBox control, including all rich text format strings in .NET 4.0?**

■ The Rtf property of the RichTextBox control is used to set or get texts, including the RTF format code.

**18. Write the steps necessary to enable a RichTextBox control to support drag and drop in .NET 4.0?**

■ First of all, set the AllowDrop property of the RichTextBox control to true. Then, add handlers for both the DragEnter and DragDrop events of the RichTextBox control, as shown in the following code snippet:

*Code for VB.*

```

AddHandler Me.RichTextBox1.DragEnter, New
    System.Windows.Forms.DragEventHandler(AddressOf
    Me.RichTextBox1_DragEnter)
AddHandler Me.RichTextBox1.DragDrop, New
    System.Windows.Forms.DragEventHandler(AddressOf
    Me.RichTextBox1_DragDrop)
Private Sub RichTextBox1_DragEnter(sender As
    Object, e As System.Windows.Forms.DragEventArgs)

```

**19. How can you set the color and font for the text displayed in the RichTextBox control in .NET 4.0?**

■ The RichTextBox class contains the Font and ForeColor properties to set the font and color of the text displayed in the RichTextBox control. The ForeColor property accepts a value from the Color class and the Font property accepts a value by initializing the Font class. The following code snippets show how to assign values to the

ForeColor and Font properties of the RichTextBox control in VB and

C#:

**Code for VB:**

```
RichTextBox1.Font = New Font("tahoma", 16)
```

```
FontStyle.Bold)
```

```
GraphicsUnit.Pixel)
```

```
RichTextBox1.ForeColor = Color.DeepSkyBlue
```

**Code for C#:**

```
RichTextBox1.Font = new Font("tahoma", 16,
```

```
FontStyle.Bold,
```

```
GraphicsUnit.Pixel)
```

```
RichTextBox1.ForeColor = Color.DeepSkyBlue;
```

**20. How can you programmatically position the cursor on a given line or on a character in the RichTextBox control in C#?**

- The RichTextBox control contains the Lines array property, which displays one item of an array in a separate line. Each line entry has a Length property, which can be used to accurately position the cursor at a character, as shown in the following code snippet:

```
private void GoToLineAndColumn(RichTextBox RTB, int
```

```
Line, int column)
{
    int offset = 0;
    for (int i = 0; i < Line - 1; i++)

```

```
    RTB.Lines[i].Length + 1);
    offset += RTB.Lines[i].Length + 1;
}
RTB.Focus();
RTB.SelectionStart = offset;
RTB.SelectionLength = column - 0;
```

**21. How can you add a hyperlink to a RichTextBox control in .NET 4.0?**

- Hyperlinks can be inserted in the RichTextBox control to redirect .NET users to any resource or perform any action, such as displaying current date and time. It is necessary to set the DetectUrls property of the RichTextBox control to True before executing the following code snippet:

**Code for VB:**

```
Private Sub RichTextBox1_LinkClicked(ByVal sender As
Object, ByVal e As
System.Windows.Forms.LinkClickedEventArgs)
    System.Diagnostics.Process.Start(e.LinkText)
End Sub
```

**22. How can you place a border around a picture box?**

- The PictureBox control offers the BorderStyle property, which can be set to define the style of its border. This property can accept any of the three values from Fixed3D, FixedSingle, or None. These properties can be easily set through code or through the Properties window of the Visual Studio IDE.

**23. How can you copy a bitmap from the clipboard to a picture box in .NET 4.0?**

- The following code snippet shows how to copy an image from clipboard to picture box:

**Code for VB:**

```
this.pictureBox1.Image =
Image.FromClipboard();
Clipboard.GetData(DataFormats
Bitmap);Clipboard.GetData(DataFormats
Bitmap);
```

**Code for C#:**

- The following code snippet demonstrates how you can drag and drop an image from one picture box (Source) to another (Target):

**Code for VB:**

```
In the Form1 load event, set AllowDrop of the Target PictureBox to true as this property cannot be set in the designer. Me.PictureBox2.AllowDrop = True
SourcePictureBox
Private Sub PictureBox1_DragLeave(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DragEventArgs)
    If e.BeyondBounds Then
        e.DragEffect = DragDropEffects.None
    End If
End Sub
TargetPictureBox
```

**Code for C#:**

```
private void Form1_Load(object sender, EventArgs e)
{
    System.Windows.Forms.PictureBox1.AllowDrop = true;
    System.Windows.Forms.PictureBox2.AllowDrop = true;
}
```

**24. How can you drag and drop an image from one picture box to another in .NET 4.0?**

- The following code snippet demonstrates how you can drag and drop an image from one picture box (Source) to another (Target):

**Code for VB:**

```
In the Form1 load event, set AllowDrop of the Target PictureBox to true as this property cannot be set in the designer. Me.PictureBox2.AllowDrop = True
SourcePictureBox
Private Sub PictureBox1_DragLeave(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DragEventArgs)
    If e.BeyondBounds Then
        e.DragEffect = DragDropEffects.None
    End If
End Sub
TargetPictureBox
```

```

    //drag drop effects
    protected void pictureBox2_DragEnter(B EventArgs sender)
    {
        DragDropEffects effects = DragDropEffects.Copy;
        if (e.Data.GetDataPresent(DataFormats.Bitmap))
        {
            effects = DragDropEffects.Move;
        }
        else
        {
            effects = DragDropEffects.None;
        }
        e.Effect = effects;
    }

    private void pictureBox2_DragDrop(B EventArgs sender)
    {
        Set the image to be the dragged image
        pictureBox2.Image = e.Data.GetData(DataFormats.Bitmap) as Bitmap;
    }
}

```

*Code for C#:*

```

//In the Form Load
//Set A11yDrop of the Target pictureBox to
//true as this property cannot be set in the
//design time
this.pictureBox2.AllowDrop = true;
//Source PictureBox
private void pictureBox1_MouseMove(object
    sender, System.Windows.Forms.MouseEventArgs e)
{
    if (e.X < 100)
    {
        pictureBox1.DragDrop(pictureBox1.Image,
            DragDropEffects.Left);
    }
}

```

```

//Target PictureBox
//Drag Drop Effects
private void pictureBox2_DragEnter(object sender)
{
    System.Windows.Forms.DragEventArgs e =
        e.Data.GetDataPresent(DataFormats.Bitmap) ?
            DragDropEffects.Copy : DragDropEffects.None;
    e.Effect = e.Effect | e.Effect;
}

```

*Code for VB:*

```

Private Sub A11yDrop(DragDropEffects effects)
    If effects = DragDropEffects.Copy Then
        effects = DragDropEffects.Move
    End If
    e.Effect = effects | e.Effect
End Sub

```

```

    this.pictureBox1.Image =
        (Bitmap)(e.Data.GetData(DataFormats.Bitmap));
}

```

**25. How can you programmatically prevent a ComboBox from dropping, in .NET 4.0?**

- To avoid dropping of a ComboBox, you need to override the WndProc() method and ignore WM\_LBUTTONDOWN and WM\_LBUTTONDBLCLK events.

**26. How can you prevent users of an application from editing the text in the ComboBox controls in .NET 4.0?**

- The ComboBox class contains the DropDownStyle property, which is used to define the display style of the items in the ComboBox control. The DropDownStyle property accepts a value from the ComboBoxStyle enumeration, which contains three members to define the styles for the items: Simple, DropDownList, and DropDown. The DropDownList value of the ComboBoxStyle enumeration is selected to set a ComboBox control as non-editable by users, as shown in the following code snippets:

*Code for VB:*

```

    comboBox1.DropDownStyle =
        ComboBoxStyle.DropDownList

```

*Code for C#:*

```

    comboBox1.DropDownStyle =
        ComboBoxStyle.DropDownList

```

**27. How can you adjust the height of a comboBox box drop-down list?**

- You can control the height of a combo box drop-down list by setting the MaxDropDownItems property of the combo box. The MaxDropDownItems property sets the maximum number of entries that will be displayed by the drop-down list.

**28. How do you set the width of a ComboBox to fit the entries in its list in C#?**

- You can iterate through the list to find the longest text extent by using the MeasureString() method and then use this as the width of a combo box, as shown in the following code snippet:

```

    System.Drawing.Graphics g =
        comboBox1.CreateGraphics();
    float maxwidth = 0f;
    foreach (Object o in comboBox1.Items)
    {
        string itemtext = o.ToString();
        float width = g.MeasureString(itemtext, comboBox1.Font).Width;
        if (width > maxwidth)
            maxwidth = width;
    }
    comboBox1.Width = maxwidth;

```

```
float width = comboBox1.Font.Width;
float maxwidth = 20; // 20 is to
// handle case of buttonwidth
```

```
g.Dispose();
comboBox1.Width = maxwidth; // 20 is to
// handle case of buttonwidth
```

### 29. How can you programmatically create a new list for the combo box drop-down list in .NET 4.0?

- The following code snippet shows the code to create a new list for the combo box drop-down list:

*Code for VB:*

```
Dim list As New DataTable()
list.Columns.Add(New DataColumn("Display",
    GetType(String)))
list.Rows.Add(1, "one")
list.Rows.Add(2, "two")
list.Rows.Add(3, "three")
list.Rows(2).Value = 3
comboBox1.DataSource = list
comboBox1.DisplayMember = "Display"
comboBox1.ValueMember = "Id";
```

*Code for C#:*

```
DataTable list = new DataTable();
list.Columns.Add(new DataColumn("Display",
    typeof(string)));
list.Columns.Add(new DataColumn("Id",
    typeof(int)));
list.Rows.Add(1, "one");
list.Rows.Add(2, "two");
list.Rows.Add(3, "three");
list.Rows[2].Value = 3;
comboBox1.DataSource = list;
comboBox1.DisplayMember = "Display";
comboBox1.ValueMember = "Id";
```

### 30. How can you drag file names from Windows Explorer and drop them in a list box in .NET 4.0?

- Place a list box on the form, set its AllowDrop property, and handle both DragEnter and DragDrop events, as shown in the following code snippet:

```
private void listBox1_DragEnter(object sender, System.Windows.Forms.DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.FileDrop))
    {
        e.Effect = DragDropEffects.All;
    }
}
```

```
private void listBox1_DragDrop(object sender, System.Windows.Forms.DragEventArgs e)
{
    string[] files = (string[])e.Data.GetData(DataFormats.FileDrop);
    foreach (string s in files)
    {
        listBox1.Items.Add(s.Substring(1 +
            lastIndexOf(@"\") + 1));
    }
}
```

*Code for VB:*

```
Private Sub listBox1_DragEnter(sender As Object, e As
    System.Windows.Forms.DragEventArgs)
    If e.Data.GetDataPresent(DataFormats.FileDrop) Then
        e.Effect = DragDropEffects.All
    Else
        e.Effect = DragDropEffects.None
    End If
End Sub
Private Sub listBox1_DragDrop(sender As Object, e As
    System.Windows.Forms.DragEventArgs)
    Dim files As String() = CType(e.Data.GetData("FileDrop"), FileDrop)
    Dim s As String
    For Each s In files
        For Each s In files
            Just filename
            listBox1.Items.Add(s.Substring(1 +
                lastIndexOf("\") + 1))
        Or full pathname
        listBox1.Items.Add(s)
    Next
    listBox1.DragDrop
End Sub
```

### 31. Write the code to set the width of a list box to fit the text in C#.

- You can iterate through the list to find the longest text extent by using the MeasureString() method, as shown in the following code snippet:

32. How can you check/uncheck all items in the **CheckedListBox** control in .NET 4.0?

- To help all its members, NET has

Code for 1/R.

Dim i a  
For i =  
myCheck  
Next

10

```
for( int i=0 ; i < myCheckedListBox.Items.Count ; i++ ) {
```

### 33. How will you pick a color from the Color Dialog box?

- To pick a color from the color dialog box, you need to create an instance of the `ColorDialog` box and invoke to the `ShowDialog()` method. The code to display the color dialog box and set the `BackColor` property of the `Label` control similar to the color selected in the color dialog box control is:

private void button1\_Click(object sender,  
EventArgs e)

```
if (colorDialog.ShowDialog() != DialogResult.Cancel)
    label1.Text = "The new color is: " + colorDialog.Color;
label1.BackColor = colorDialog.Color;
```

34. How do you use the **FontDialog** class to set a control's font in C#?

- To use the `FontDialog` class to set a control's font, create an instance of the class and call its `ShowDialog()` method, as shown in the following code snippet:

```
FontDialog1 = new FontDialog();
FontDialog1.ShowColor = true;
FontDialog1.ShowDialog();
if (FontDialog1.ShowDialog() == DialogResult.Cancel)
    return;
textBox1.Font = FontDialog1.Font;
textBox1.ForeColor = FontDialog1.Color;
}
```

35. Write a method to get only the name of a file from the complete path.

- Use a FileInfo class and instantiate its object with the full path as the constructor argument and then simply call the FileInfo.Name file and you will get just the name of the file

control in .NET 4.0?

The following code snippet is used to add subitems to a ListView control in C#?

```

    List<ViewItem> item = new List<ViewItem>();
    item.AddRange(new ViewItem[] { item1, item2 });
    item.Add(item3);
    item.AddRange(new ViewItem[] { item4, item5 });
    string[] itemNames = item.Select(x => x.Name).ToArray();

```

37. How can you unselect the selected items in a ListView control

- The syntax to unselect the selected items in the List/View control is shown in the following code snippets:

To pick a color from the color dialog box, you need to create an instance of the `ColorDialog` box and invoke its `ShowDialog` method.

button1.Click[object sender];

8. Write a code to select an item in the ListView control programmatically in C#.

- To select an item from the ListView control, you can use the following

The *Journal of Clinical Endocrinology* is published monthly by Blackwell Scientific Press.

9. How can we check whether a child form is already displayed or not so that we don't display it again?

- You can check whether a child form is already displayed or not by using the following code snippet:

```
// MyChildForm is the one I'm looking for
MyChildForm childForm = null;
foreach (Form f in this.MdiChildren)
{
    if (f is MyChildForm)
    {
        found = true;
        childForm = (MyChildForm)f;
        break;
    }
}
if (childForm != null)
{
    childForm.Show();
    childForm.Focus();
}
```

**40. We need to perform certain custom processing whenever an MDI child form is added /removed from the MDIClient form. How do we achieve this in C#?**

- The MDIClient forms have an MDIClient child window and it is to this MDIClient window that MDI child forms are parented. The MDIClient's ControlAdded/ControlRemoved event gets fired whenever a child form is added or removed. You can subscribe to these events and add the required processing code from within the handlers, as shown in the following code snippet:

```
// From within the MDIClient form, subscribe to
the MDIClient's
ControlAdded/ControlRemoved events
foreach(Control ctrl in this.Controls)
{
    if(ctrl.GetType() == typeof(MdiClient))
    {
        ctrl.ControlAdded += new
        ControlEventHandler(this.MdIClient_ControlAdded);
        ctrl.ControlRemoved += new
        ControlEventHandler(this.MdIClient_ControlRemoved);
    }
}
```

**42. How does an MDI form differ from a standard form?**

- An MDI form closely resembles a standard form with one major difference—the client area of an MDI form acts as a container for other forms. It means that an MDI form, also known as an MDI parent form, can display MDI child forms inside it.

**43. Can you write a class without specifying a namespace? Which**

- namespace does it belong to by default?
- Yes, we can write a class without specifying namespace and that class belongs to a global namespace that has no name.

**41. How can you make your child form fill the entire MDI client without getting maximized in .NET 4.0?**

- Doing this involves all docked controls (including menus) in the MDI parent form. The code to perform this activity is given in the following code snippet:

*Code for VB:*

```
Private Sub FillActiveChildFormToMDIClient()
    Dim child As Form = Me.ActiveMdiChild
    Dim mdiClientArea As Rectangle
    Rectangle.Empty
    Dim c As Control
    For Each c In Me.Controls
        If TypeOf c Is MDIClient Then
            mdiClientArea = c.ClientRectangle
        End If
    Next
    child.Bounds = mdiClientArea
End Sub
```

*Code for C#:*

```
private void FillActiveChildFormToMdiClient()
{
    Form child = this.ActiveMdiChild;
    Rectangle mdiClientArea = Rectangle.Empty;
    foreach (Control c in this.Controls)
    {
        if (c is MdiClient)
        {
            mdiClientArea = c.ClientRectangle;
        }
    }
    child.Bounds = mdiClientArea;
}
```

**44. What are the three states set in the CheckState property of Checkbox?**

- Checked
- Unchecked
- Indeterminate

**45. What is the difference between the Add() and Insert() methods of a ListBox control?**

- The Add() method simply adds an item into the list box; whereas, the Insert() method inserts an item at the specified index.

**46. Where does an ImageList control appear when you add it at the design time?**

- The ImageList control is a component; therefore, it appears in the component tray at the design time.

**47. Consider a situation where you have added panels in a StatusBar control; however, they are not displayed at run time. What could be the reason for this?**

- To display panels in the StatusBar control, the ShowPanels property needs to be set to true.

**48. Does a Timer control appear at run time?**

- Timer is a component; therefore, it does not appear at run time.

**49. What is the difference between a ToolStrip control and a ContextMenuStrip control?**

- The difference between a ToolStrip control and a ContextMenuStrip control is that a ToolStrip control is associated with the Windows Form; whereas, a ContextMenuStrip control is associated with a control, which is added to the Windows Form.

**50. What is the difference between a toolStrip drop-down button and a toolStrip split button?**

- The difference between a toolStrip drop-down button and a toolStrip split button is that a toolStrip split button is a combination of two controls—a push button and a drop-down button; whereas, a toolStrip drop-down button is a single control.

**51. What is the use of a toolStrip container?**

- A toolStrip container is used to contain controls, such as ToolStrip, MenuStrip, and StatusStrip, so that these controls can be docked and moved at the run time.

**52. Is it possible to associate a control with more than one ContextMenu?**

- No, we cannot associate a control with more than one ContextMenu.

**53. What is the difference between a CheckBox control and a RadioButton control?**

- A CheckBox control is square shaped; whereas, a RadioButton control is round in shape. Moreover, you can select more than one CheckBox control from a group of CheckBox controls; whereas, you can select only a single RadioButton control from a group of RadioButton controls.

**54. Why do you require user-defined controls?**

- User-defined controls are particularly useful in situations where you need to enhance the functionality of an existing control.

**55. What is the difference between a ListBox control and a ComboBox control?**

- With a ListBox control, the user can only make a selection from a list of items; whereas, with a ComboBox control, the user can make a selection from the list of items as well as can add custom entry and select the same.

**56. What is the difference between the Panel and GroupBox control?**

- The Panel and GroupBox controls both can be used as a container for other controls, such as radio buttons and check box. The main differences between a Panel and a GroupBox control are as follows:
  - Panel does not display captions, while GroupBox does
  - Panel has scrollbar, while GroupBox does not

**57. What is the importance of a Button control?**

- A Button control is an important Windows control, which provides the most common way of creating and handling an event in the code with the help of its Click event.

**58. What are the values that can be assigned to the DialogResult property of a Button control?**

- The DialogResult property of a Button control can be assigned a value from the DialogResult enumerations, which are as follows:
  - Abort-Returns Abort
  - Cancel-Returns Cancel

- Ignore>Returns Ignore
- No>Returns No
- None>Nothing is returned from the dialog box
- OK>Returns OK
- Retry>Returns Retry
- Yes>Returns Yes

#### 59. How can you move and resize a control on a Windows form?

- You can make use of the SetBounds() method to move as well as resize the control on a Windows form.

#### 60. When would you use the ErrorProvider control? Provide an example of using the ErrorProvider control in C#.

- The ErrorProvider control is used to provide validations to Windows application and display user-friendly messages to the user, if the validation fails. For example, if you want to validate textBox1, you need to place the ErrorProvider control on the form and then add the following code in the Code window:

```
private void textBox1_Validating (object sender,
System.ComponentModel.CancelEventArgs e)
{
    ValidateName();
}

private bool ValidateName()
{
    bool bstatus = true;
    if (textBox1.Text == "")
    {
        errorProvider1.SetError(
            textBox1, "Please enter your Name.");
        bstatus = false;
    }
    else
        errorProvider1.SetError(textBox1, "");
}
return bstatus;
```

In this code snippet, the ErrorProvider control checks whether textBox1 is empty or not. If it is empty, then a user defined message is displayed.

#### 61. How would you create an ellipse, which is a non-rectangular window?

- Open a new Windows form, which is by default rectangular in design and then set the TransparencyKey property to the same value as BackColor, which will effectively make the background of the form

transparent. Then, set the FormBorderStyle property to FormBorderStyle.None, which removes the contour and contents of the form.

#### 62. How do you create a separator in the Menu Designer?

- You can use hyphen (-) to create a separator.

#### 63. What is the difference between the WindowsDefaultLocation and WindowsDefaultBounds properties?

- The WindowsDefaultLocation property makes the form to start up at a location selected by the operating system, but with internally specified size. The WindowsDefaultBounds property delegates both size and starting position choices to the operating system.

#### 64. What is the difference between pixels, points, and em's when fonts are displayed?

- A pixel is the lowest-resolution dot that the computer monitor supports. Its size depends on user's settings and the size of the monitor. A point is always 1/72 of an inch. An em is the number of pixels it takes to display the letter M.

#### 65. How do you retrieve the customized properties of a .NET application from the XML.config file?

- Initialize an instance of the AppSettingsReader class. Call the GetValue method of the AppSettingsReader class, passing in the name of the property and the type expected. Finally, assign the result to the appropriate variable.

#### 66. How can you display a default value in the text box of an input box?

- You can display a default value in the text box of an input box by using the DefaultResponse argument of the InputBox() function.

#### 67. Name the parent class for all Windows controls.

- The Control class or System.Windows.Forms.Control class is the parent class for all Window controls.

#### 68. How can you add a Button control dynamically at runtime in C#?

- You can add a Button control dynamically at runtime by writing the following code:

```
using System;
using System.Drawing;
```

**Using System.Windows.Forms; class AddControl : Form;**

```
Button bs = new Button();
public AddControl()
{
    InitializeComponent();
    bs.Click += new EventHandler(bs_Click);
}
private void button1_Click(object sender, EventArgs e)
{
    bs.Text = "Hello";
    bs.Location = new Point(50, 50);
    bs.Size = new Size(50, 50);
    this.Controls.Add(bs);
}
private void bs_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello");
}
```

**72. What is the MaskedTextBox control? What does the Mask property do?**

- The MaskedTextBox control is an improvement of the TextBox control. It forces the user to provide the proper input, which is specified by the Mask property. In other words, it prevents the user to provide any invalid input to an application. The Mask property gets or sets the input type to the MaskedTextBox control. There are many built-in formats for the Mask property, such as phone no., short date, time, zip code, and custom.

**73. Is it possible to add an image on the RadioButton control?**

- Yes, you can add an image on the RadioButton control by setting the Image property.

**74. Create an application in C# to accept user name and password. The application must check the user name and password whether it is valid or not. Application will provide three login attempts to a user. If the user does not succeed to login, then the application should terminate.**

- The code to create this application is given in the following code snippet:

```
using System.Drawing;
using System.Text;
using System.Windows.Forms;
public class Login : Form
{
    string name;
    string pwd;
    int ctr = 0;
    public Form1()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        name = textBox1.Text;
        pwd = textBox2.Text;
        if (ctr < 3)
        {
            if (name == "sa" &amp; pwd == "123")
            {
                MessageBox.Show("Successfull");
            }
            else
            {
                MessageBox.Show("Incorrect
                    password");
            }
        }
        else
        {
            MessageBox.Show("You have exceeded
                the maximum number of attempts.");
        }
    }
}
```

**69. What does the DialogResult property of a Button control do?**

- The DialogResult property retrieves or sets a value that is returned to the parent form when the button is clicked.

**70. Is it possible to enter more than one line in a TextBox control?**

- Yes, it is possible to enter more than one line in a TextBox control. To do this, you need to set the Multiline property of the TextBox control to True. You can set this property at design time as well as runtime. The syntax to set this property at runtime is as follows:

```
RichTextBox1.Multiline = true;
```

**71. Differentiate between a TextBox control and Rich TextBox control.**

- The TextBox control is an input control, which allows a user to enter text to an application at runtime. By default, it allows only single line text; however, you can change its property to accept the multiline text as well as scroll bar also.

The Rich TextBox control is similar to the TextBox control with the difference that it allows the user to format its text also. You can format the text in various ways, such as bold, italic, and underlined as well as change its color and font. You can save your Rich TextBox value to a RTF (Rich Text Format) file and load value of RTF file to the Rich TextBox control.



**85. How can you add a new tab page in a Tab control in C#?**

- The code to add a new tab page in a Tab control is shown in the following code snippet:

```
private void button1_Click(object sender, EventArgs e)
{
    TabPage tabPage = new TabPage();
    tabPage.Text = "tabPage3";
    tabControl1.TabPages.Add(tabPage);
}
```

**86. Define the TrackBar control.**

- The TrackBar control, also known as the slider control, works as a navigator to display a large amount of information or for visual adjustment of numeric setting. There are two parts in a TrackBar control—thumb (also known as slider) and tick marks. The thumb part acts as a slider. You can adjust the thumb part using the Value property. The tick marks are visual indicators that are spaced at regular intervals.

**87. What does the TickFrequency property of the TrackBar control do?**

- The TickFrequency property gets or sets a value that specifies the distance between ticks. By default, the distance between ticks is 1.

**88. Describe the ToolTip control. How can you associate it with other controls?**

- The ToolTip control generates a small pop-up window with explanatory text for an element. It is displayed when the user pauses the mouse for a certain period over an element/control. Tool tips provide a quick help to user to understand about that element. To associate a tool tip with other control, you need to implement the SetToolTip() method.

**89. What is the function of MinDate and MaxDate properties of the MonthCalendar control?**

- The MinDate and MaxDate properties allow users to get and set the minimum and maximum allowable date.

**90. What does the Checked property of the DateTimePicker control do?**

- The Checked property holds either true or false value. It holds true, when the Value property hold a valid date-time value and is updatable; otherwise, false.

**91. What is the use of a Timer control? Can a Timer control pause?**

- The Timer control is a mechanism to perform an iterative task at a specified time interval. You cannot pause it because it can only start and stop.

**92. How can you get or set the time between Timer ticks?**

- There is an Interval property, which is responsible to get and set the time in milliseconds.

**93. What does the PerformStep() method do?**

- The PerformStep() method increases the value of Progress bar according to the amount set by the Step property.

**94. Which class manages the event and layout of all ToolStrip elements?**

- The ToolStripItem class manages the event and layout of all elements that the ToolStrip control contains.

**95. Name the classes used to handle standard menu in a ToolStrip control.**

- The two main classes used to handle standard menu in a ToolStrip control are:

- MenuStrip* —Acts as a container for the menu structure of a form.
- ToolStripMenuItem* —Supports the items in a menu system (including the menus, such as File and Edit).

**96. Is it possible to associate a control with more than one ContextMenu control?**

- No, we cannot associate a control with more than one ContextMenu control.

**97. What is a StatusStrip control?**

- The StatusStrip control represents an area at the bottom of a Windows Form where an application can display various kinds of status information. You can use this control for performing tasks, such as showing the number of words in a document or the progress of a printing task. Typically, the StatusStrip control contains ToolStripStatusLabel objects, which display text, an icon, or both. By default, the StatusStrip control has no panels. You can add panels to it using the ToolStripItemCollection.AddRange method or use the StatusStrip Items Collection Editor dialog box at design time to add panels. The StatusStrip control provides the following controls, which are used for displaying any type of status information in an application:

- ToolStripStatusLabel* control—Displays messages
- ToolStripProgressBar* control—Displays the progress of a background task, such as saving the document or sending it to the printer

- ToolStripDropDownButton* control—Displays a drop-down list
- ToolStripSplitButton* control—Represents a combination of a standard button on the left and a drop-down button on the right

the printer

## 98. How can you display an icon at runtime on the StatusStrip control?

- The following code snippet shows the code to display an icon at runtime on the StatusStrip control:

```
toolStripStatusLabel2.Image = 
    Bitmap.FromFile("D:\Books\1.bmp");
```

## 99. Which method provides the functionality to display a dialog box at runtime?

- The ShowDialog() method is used to display the dialog box at run time.

## 100. What does the OpenFileDialog method of the OpenFileDialog control do?

- The OpenFileDialog() method opens the file selected by the user with read-only permission. The file is specified by the FileName property.

## 101. What is a PrintPreview control? Give the function of the Document property of the PrintPreview control?

- The PrintPreviewControl control displays a document to be printed, as it will appear when printed, that is it displays a preview of the document to be printed. This control has no buttons or any other user interface elements. This control is typically used only when there is a need to write custom print-preview user interfaces. To have the standard user interface for print preview, the PrintPreviewDialog control is used. You can use PrintPreviewDialog objects to create your own custom print previews. The Document property gets or sets the value that indicates the document to preview.

## 3. What are the benefits of using of ADO.NET in .NET 4.0.

- The following are the benefits of using ADO.NET in .NET 4.0 are as follows:

- Language-Integrated Query (LINQ)*—Adds native data-querying capabilities to .NET languages by using a syntax similar to that of SQL. This means that LINQ simplifies querying by eliminating the need to use a separate query language. LINQ is an innovative technology that was introduced in .NET Framework 3.5.
- 102. How many parameters are required to the *ErrorMessage* method?**
- The ErrorMessage() method is used to set the description of the string that is displayed when an error occurs for a specified control. This method takes two parameters. First is the control name and the second is the message, which you want to display.

# 5

## ADO.NET

1. Mention the full form of ADO.

- The full form of ADO is ActiveX Data Object.

2. Explain ADO.NET in brief.

- ADO.NET is a very important feature of .NET Framework, which is used to work with data that is stored in structured data sources, such as databases and XML files. The following are some of the important features of ADO.NET:

- Contains a number of classes that provide you with various methods and attributes to manage the communication between your application and data source.
- Enables you to access different data sources, such as Microsoft SQL Server, and XML, as per your requirements.
- Provides a rich set of features, such as connection and commands that can be used to develop robust and highly efficient data services in .NET applications.
- Provides various data providers that are specific to databases produced by various vendors. For example, ADO.NET has a separate provider to access data from Oracle databases; whereas, another provider is used to access data from SQL databases.

- **LINQ to DataSet**—Allows you to implement LINQ queries for disconnected data stored in a dataset. LINQ to DataSet enables you to query data that is cached in a DataSet object. DataSet objects allow you to use a copy of the data stored in the tables of a database, without actually getting connected to the database.

- **LINQ to SQL**—Allows you to create queries for data stored in SQL server database in your .NET application. You can use the LINQ to SQL technology to translate a query into a SQL query and then use it to retrieve or manipulate data contained in tables of an SQL Server database. LINQ to SQL supports all the key functions that you like to perform while working with SQL, that is, you can insert, update, and delete information from a table.
- **SqlClient Support for SQL Server 2008**—Specifies that with the starting of .NET Framework version 3.5 Service Pack (SP) 1, .NET Framework Data Provider for SQL Server (System.Data.SqlClient namespace) includes all the new features that make it fully compatible with SQL Server 2008 Database Engine.
- **ADO.NET Data Platform**—Specifies that with the release of .NET Framework 3.5 Service Pack (SP) 1, an Entity Framework 3.5 was introduced that provides a set of Entity Data Model (EDM) functions. These functions are supported by all the data providers; thereby, reducing the amount of coding and maintenance in your application. In .NET Framework 4.0, many new functions, such as string, aggregate, mathematical, and date/time functions have been added.

#### 4. Explain the architecture of ADO.NET in brief.

- ADO.NET consists of two fundamental components:
- The DataSet, which is disconnected from the data source and does not need to know where the data that it holds is retrieved from.
- The .NET data provider, which allows you to connect your application to the data source and execute the SQL commands against it.

The data provider contains the Connection, Command, DataReader, and DataAdapter objects. The Connection object provides connectivity to the database. The Command object provides access to database commands to retrieve and manipulate data in a database. The DataReader object retrieves data from the database in the read-only and forward-only mode. The DataAdapter object uses Command objects to execute SQL commands. The DataAdapter object loads the

#### 5. Mention different types of data providers available in .NET Framework.

- Table 5.1 lists the data providers available in .NET Framework:

Provider	Description
.NET Framework Data Provider for SQL Server	Provides access to Microsoft SQL Server 7.0 or later version. It uses the <b>System.Data.SqlClient</b> namespace.
.NET Framework Data Provider for OLE DB	Provides access to databases exposed by using OLE DB. It uses the <b>System.Data.OleDb</b> namespace.
.NET Framework Data Provider for ODBC	Provides access to databases exposed by using ODBC. It uses the <b>System.Data.Odbc</b> namespace.
.NET Framework Data Provider for Oracle	Provides access to Oracle database 8.1.7 or later versions. It uses the <b>System.Data.OracleClient</b> namespace.

#### 6. Explain in brief DataAdapter class in ADO.NET.

- The DataAdapter class retrieves data from the database, stores data in a dataset, and reflects the changes made in the dataset to the database. The DataAdapter class acts as an intermediary for all the communication between the database and the Dataset object. The DataAdapter class is used to fill a DataTable or DataSet object with data from the database using the `Fill()` method. The DataAdapter class applies the changes made in dataset to the database by calling the `Update()` method. The DataAdapter class provides four properties that represent the database command: `SelectCommand`, `InsertCommand`, `DeleteCommand`, and `UpdateCommand`.

#### 7. What is a DataReader object?

- The DataReader object helps in retrieving the data from a database in a forward-only, read-only mode. The base class for all the DataReader objects is the **DbDataReader** class. The **DataReader** object is returned as a result of calling the `ExecuteReader()` method of the **Command** object. The DataReader object enables faster retrieval of data from databases and enhances the performance of .NET applications by providing rapid data access speed. However, it is less preferred as compared to

the DataAdapter object because the DataReader object needs an Open connection till it completes reading all the rows of the specified table. An Open connection to read data from large tables consumes most of the system resources. When multiple client applications simultaneously access a database by using the DataReader object, the performance of data retrieval and other related processes is substantially reduced. In such a case, the database might refuse connections to other .NET applications until other clients free the resources.

#### 8. What is the role of the DataSet object in ADO.NET?

- One of the major component of ADO.NET is the DataSet object, which always remains disconnected from the database and reduces the load on the database.

#### 9. Describe the disconnected architecture of ADO.NET's data access model.

- ADO.NET maintains a disconnected database access model, which means, the application never remains connected constantly to the data source. Any changes and operations done on the data are saved in a local copy (dataset) that acts as a data source. Whenever, the connection to the server is re-established, these changes are sent back to the server, in which these changes are saved in the actual database or data source.

#### 10. Mention the namespace that is used to include .NET Data Provider for SQL server in .NET code.

- The System.Data.SqlClient namespace.

#### 11. What are the usages of the Command object in ADO.NET?

- The following are the usages of the Command object in ADO.NET:
  - The Command object in ADO.NET executes a command against the database and retrieves a DataReader or Dataset object.
  - It also executes the INSERT, UPDATE, or DELETE command against the database.
  - All the Command objects are derived from the DbCommand class.
  - The Command object is represented by two classes: SqlCommand and OleDbCommand.
  - The Command object provides three methods to execute commands on the database:
    - The ExecuteNonQuery() method executes the commands and does not return any value
    - The ExecuteScalar() method returns a single value from a database query
    - The ExecuteReader() method returns a result set by using the DataReader object

#### 12. What are the various methods provided by the DataSet object to generate XML?

- The various methods provided by the DataSet object to generate XML are:

- ReadXml() —Reads XML document into a DataSet object
- GetXml() —Returns a string containing an XML document
- WriteXml() —Writes an XML data to disk

#### 13. How can you identify whether or not any changes are made to the DataSet object since it was last loaded?

- The DataSet object provides the following two methods to track down the changes:
    - The GetChanges method—Returns the DataSet object, which is changed since it was loaded or since the AcceptChanges method was executed.
    - The HasChanges method—Indicates if any changes occurred since the DataSet object was loaded or after a call to the AcceptChanges method was made.
- If you want to revert all changes since the DataSet object was loaded, use the RejectChanges method.

#### 14. How can you add or remove rows from the DataTable object of DataSet?

- The DataRowCollection class defines the collection of rows for the DataTable object in a dataset. The DataTable class provides the NewRow method to add a new DataRow to DataTable. The NewRow method creates a new row, which implements the same schema as applied to the DataTable. The following are the methods provided by the DataRowCollection object:
  - Add()—Adds a new row to DataRowCollection
  - RemoveAt()—Removes a DataRow object from DataRowCollection index number

**15. What is the use of DataView?**

- User-defined view of a table is contained in a DataView. A complete table or a small section of table depending on some criteria can be presented by an object of the DataView class. You can use this class to sort and find data within DataTable.

DataView has the following methods:

- `Find()`—Finds a row in a DataView by using sort key value.
- `FindRows()`—Uses the sort key value to match it with the columns of DataRowView objects. It returns an array of all the corresponding objects of DataRowView whose columns match with the sort key value.
- `AddNew()`—Adds a new row to the DataView object.
- `Delete()`—Deletes the specified row from the DataView object according to the specified index.

**16. What is the use of the CommandBuilder class?**

- The CommandBuilder class is used to automatically update a database according to the changes made in a DataSet. This class automatically registers itself as an event listener to the RowUpdating event. Whenever data inside a row changes, the object of the CommandBuilder class automatically generates an SQL statement and uses the SelectCommand property to commit the changes made in DataSet.

OLEDB provider in .NET Framework has the `OleDbCommandBuilder` class; whereas, the SQL provider has the `SqlCommandBuilder` class.

**17. What is the meaning of object pooling?**

- Object pooling is a concept of storing a pool (group) of objects in memory that can be reused later as needed. Whenever, a new object is required to create, an object from the pool can be allocated for this request; thereby, minimizing the object creation. A pool can also refer to a group of connections and threads. Pooling, therefore, helps in minimizing the use of system resources, improves system scalability, and performance.

**18. What is the use of the connection object?**

- The connection object is used to connect your application to a specific data source by providing the required authentication information in connection string. The connection object is used according to the type of the data source. For example, the

`OleDbConnection` object is used with an OLE-DB provider and the `SqlConnection` object is used with an MS SQL Server.

**19. What is connection pooling?**

- Connection pooling refers to the task of grouping database connections in cache to make them reusable because opening new connections every time to a database is a time-consuming process. Therefore, connection pooling enables you to reuse already existing and active database connections, whenever required, and increasing the performance of your application.
- You can enable or disable connection pooling in your application by setting the `Pooling` property to either true or false in connection string. By default, it is enabled in an application.

**20. What are the parameters that control most of connection pooling behaviors?**

- The parameters that control most of connection pooling behaviors are as follows:
  - Connect Timeout
  - Max Pool Size
  - Min Pool Size
  - Pooling

**21. What are the pre-requisites for connection pooling?**

- The prerequisites for connection pooling are as follows:
  - There must be multiple processes to share the same connection describing the same parameters and security settings.
  - The connection string must be identical.

**22. Explain the DataAdapter.Update() and DataSet.AcceptChanges() methods.**

- The `DataAdapter.Update()` method calls any of the DML statements, such as the UPDATE, INSERT, or DELETE statements, as the case may be to update, insert, or delete a row in a DataSet. The `DataSet.AcceptChanges` method reflects all the changes made to the row since the last time the `AcceptChanges` method was called.

**23. What property must be set and what method must be called in your code to bind the data from some data source to the Repeater control?**

- You must set the `DataSource` property and call the `.DataBind()` method.

- 24. Name the method that needs to be invoked on the DataAdapter control to fill the generated DataSet with data?**
- The Fill() method is used to fill the dataset with data.
- 25. Name the two properties of the GridView control that have to be specified to turn on sorting and paging.**
- The properties of the GridView control that need to be specified to turn on sorting and paging are as follows:
    - The AllowSorting property of the GridView control indicates whether sorting is enabled or not. You should set the AllowSorting property to true to enable sorting.
    - The AllowPaging property of the GridView control indicates whether paging is enabled or not. You should set the AllowPaging property to true to enable paging.
- 26. Which namespaces are required to enable the use of databases in ASP.NET pages?**
- The following namespaces are required to enable the use of databases in ASP.NET pages:
    - The System.Data namespace
    - The System.Data.OleDb namespace (to use any data provider, such as Access, Oracle, or SQL)
    - The System.Data.SqlClient namespace (specifically to use SQL as the data provider)
- 27. Which object is used to add a relationship between two DataTable objects?**
- The DataRelation object is used to add relationship between two DataTable objects.
- 28. What are different types of authentication techniques that are used in connection strings to connect .NET applications with Microsoft SQL Server?**
- .NET applications can use two different techniques to authenticate and connect with SQL Server. These techniques are as follows:
    - The Windows Authentication option
    - The SQL Server Authentication option
- 29. Out of Windows authentication and SQL Server authentication, which authentication technique is considered as a trusted authentication method?**
- The Windows authentication technique is considered as a trusted authentication method because the username and password are checked with the Windows credentials stored in the Active Directory. The SQL Server Authentication technique is not trusted as all the values are verified by SQL Server only.
- 30. How can you implement transactions in ADO.NET?**
- The following are the most common sequence of steps that would be performed while implementing transactions in ADO.NET:
    1. Make a connection to the database
    2. Create the SqlCommand object
    3. Open the database connection by using the open() method of the **connection** object
    4. Call the BeginTransaction() method of the **connection** object to start a transaction that generates the **transaction** object.
    5. Execute the SQL commands using the **command** object
    6. Call the Commit() method of the **transaction** object to complete the transaction and the Rollback() method to abort the transaction
    7. Close the database connection by using the Close() method
- 31. How would you connect to a database by using .NET?**
- The Connection class is used to connect a .NET application with a database.
- 32. What is the difference between OLEDB Provider and SqlConnection?**
- With respect to usage, there is no difference between OLEDB Provider and SqlConnection. The difference lies in their performance. SqlConnection is explicitly used to connect your application to SQL server directly. OLEDB Provider is generic for various databases, such as Oracle and Access including SQL Server. Therefore, there will be an overhead which leads to performance degradation.

**33. Which property is used to check whether a DataReader is closed or opened?**

- The `IsClosed` property is used to check whether a DataReader is closed or opened. This property returns a true value if a Data Reader is closed, otherwise a false value is returned.

**34. What is the difference between the `Clone()` and `Copy()` methods of the `DataSet` class?**

- The `Clone()` method copies only the structure of a `DataSet`. The copied structure includes all the relation, constraint, and `DataTable` schemas used by the `DataSet`. The `Clone` method does not copy the data, which is stored in the `DataSet`. The `Copy()` method copies the structure as well as the data stored in the `DataSet`.

**35. Which architecture does Datasets follow?**

- Datasets follow the disconnected data architecture.

**36. Which adapter should you use, if you want to get the data from an Access database?**

- `OleDbDataAdapter` is used to get the data from an Access database.

**37. Which properties are used to bind a `DataGridView` control?**

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

**38. Explain about ADO.NET Entity Framework in brief.**

- Entity Framework of ADO.NET allows you to focus on data through an object model instead of a relational model. Entity Framework allows you to write less data access code, reduce maintenance, summarize the structure of data into a user friendly one, and facilitate the persistence of data. You can use a data provider called `EntityClient` and a language called `Entity SQL` to interact with the entity framework directly. This provider and language is a new feature of .NET introduced by .NET 3.5. EntityClient uses the `EntityConnection` and `EntityCommand` objects to return a `DbDataReader`. You can also employ Object Services by using the `ObjectQuery` object with `Entity SQL` or using `LINQ to Entities`. You can see the overview of the ADO.NET Entity Framework in Figure 5.1:

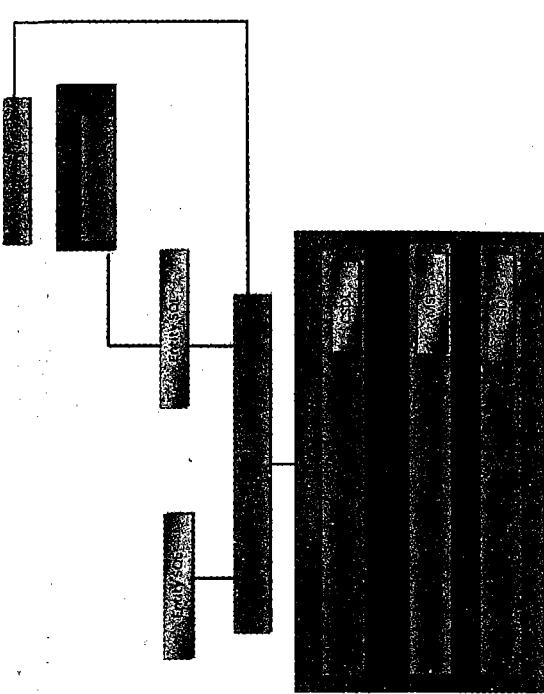


Figure 5.1: ADO.NET Entity Framework

Entity Framework supports a logical store model that represents a relational schema from a database. The Entity Framework bridges the gap between the models using a mapping layer. The layers available in Entity Framework are: Conceptual layer, Mapping layer, and Logical layer. These three layers help in mapping the data from a relational database to a more object-oriented business model. The Entity Framework provides the Conceptual Schema Definition Language (CSDL) to define the Conceptual model in an XML file. The CSDL defines the entities and relationships as defined by the application business layer. The Logical model represents the database schema. The Store Schema Definition Language (SSDL) is used to define the Logical model in XML. The Conceptual model and the Logical model associate the entities one-to-one. The Mapping layer is defined by using the Mapping Schema Language (MSL). It maps the other two layers between each other.

Entity Framework includes the following components:

- **Entity Data Model (EDM)**—Used by Entity Framework to define the conceptual entities that can be read in a serialized form using a `DataReader`.

- 6**
- *Entity SQL*—Defines a common SQL based query language which is extended to express queries in terms of EDM concepts. It is supported by both the EntityClient and Object Services.
  - *EntityClient*—An ADO.NET data provider, which exposes data in terms of conceptual EDM, which is queried through a common Entity SQL language.
  - *Object Services*—Allows you to interact with a conceptual model through a set of CLR classes. They also provide support for Entity Framework by providing services, such as state management, identity resolution, and query building support for Entity SQL.
- 
- 3.9. Explain the new features in ADO.NET Entity Framework 4.0.**
- ADO.NET Entity Framework 4.0 is introduced in .NET Framework 4.0 and includes the following new features:
  - *Persistence Ignorance*—Facilitates you to define your own Plain Old CLR Objects (POCO) which are independent of any specific persistence technology.
  - *Deferred or Lazy Loading*—Specifies that related entities can be loaded automatically whenever required. You can enable lazy loading in your application by setting the DeferredLoadingEnabled property to **true**.
  - *Self-Tracking Entities*—Refers to the entities that are able to track their own changes. These changes can be passed across process boundaries and saved to the database.
  - *Model-First Development*—Allows you to create your own EDM and then generate relational model (database) from that EDM with matching tables and relations.
  - *Built-in Functions*—Enables you to use built-in SQL Server functions directly in your queries.
  - *Model-Defined Functions*—Enables you to use the functions that are defined in conceptual schema definition language (CSDL).
- 
- 4. Write the basic syntax of a LINQ query in Visual Basic as well as in C#.**
- In Visual Basic, the basic syntax of a LINQ query starts with the From clause and ends with the Select or Group By clause. In addition, you can use the Where, Order By, and Order By Descending clauses to perform additional functions, such as filtering data and generating the data in a specific order.

In C#, the basic syntax of a LINQ query starts with the **From** clause and ends with the **Select** or **group by** clause. In addition, you can use the **where**, **orderby**, and **Orderby descending** clauses to perform additional functions, such as filtering data and generating the data in a specific order.

#### 5. In which statement the LINQ query is executed?

- A LINQ query is executed in the **For Each** statement in Visual Basic and in the **foreach** statement in C#.

#### 6. What are standard query operators in LINQ?

- The standard query operators in LINQ are the extension methods that form the LINQ pattern. These operators form an API that enables querying of any .NET array or collection. It operates on sequences and allows you to perform operations, such as determining if a value exists in the sequence and performing an aggregated function, such as a summation over a sequence.

#### 7. Which interface implements the standard query operators in LINQ?

- The standard query operators implement the **IEnumerable<T>** or the **IQueryable<T>** interface in C# and the **IEnumerable(Of T)** or the **IQueryable(Of T)** interface in Visual Basic.

#### 8. List the various standard query operators based on their functionality.

- The various standard query operators in LINQ are listed in Table 6.1.

Table 6.1: Standard Query Operators

Operator	Description
Sorting operator	Changes the order of elements of sequence returned by the query.
Filtering operator	Restricts the result set to contain those elements that satisfy a specific condition that are returned.
Join operator	Combines two collections into one.
Group operator	Divides the input sequence into groups.
Generation operator	Creates a new sequence of values.
Conversion operator	Converts a collection to an array.
Zip operator	Combines two collections into one.

#### 9. What is the function of the Distinct clause in a LINQ query?

- The Distinct clause returns the result set without the duplicate values.

#### 10. Which among these is used as a filtering operator?

- a. Where
- b. Contains
- c. Any
- d. ToListLookup

#### 11. What is the difference between the Take and Skip clauses?

- The **Take** clause returns a specified number of elements. For example, you can use the **Take** clause to return two values from an array of numbers. The **Skip** clause skips the specified number of elements in the query and returns the rest. For example, you can use the **Skip** clause to skip the first four strings in an array of strings and return the remaining array of string.

#### 12. What type of join does the Join clause perform?

- a. Inner Join

Operator	Description
Projection operator	Transforms an object into a different type of new object.
Partitioning operator	Divides the input sequence into sections.
Join operator	Combines collections.
Group operator	Creates groups.
Generation operator	Creates a new sequence of values.
Conversion operator	Converts a collection to an array.
Zip operator	Combines two collections into one.

**b. Outer Join**

- a. The Join clause performs the Inner join.

**13. On what parameter does the GroupBy clause group the data?**

- The **GroupBy** clause groups the elements that share a common attribute.

**14. What is the difference between the Select clause and SelectMany() method in LINQ?**

- Both the **Select** clause and **SelectMany()** method are used to produce a result value from a source of values. The difference lies in the result set. The **Select** clause is used to produce one result value for every source value. The result value is a collection that has the same number of elements from the query. In contrast, the **SelectMany()** method produces a single result that contains a concatenated collection from the query.

**15. What are the different implementations of LINQ?**

- The different implementations of LINQ are:
  - **LINQ to SQL**—Refers to a component of .NET Framework version 3.5 that provides a run-time infrastructure to manage relational data as objects
  - **LINQ to DataSet**—Refers to a component that makes it easier and faster to query over data cached in a **DataSet** object
  - **LINQ to XML**—Provides an in-memory XML programming interface
  - **LINQ to Objects**—Refers to the use of LINQ queries with any **IEnumerable** or **IEnumerable<T>** collection directly, without the use of an intermediate LINQ provider or API, such as LINQ to SQL or LINQ to XML

**16. What is Object Relational Designer (O/R Designer)?**

- The O/R Designer provides a visual design surface to create LINQ to SQL entity classes and associations (relationships) that are based on objects in a database.

**17. How can you open the O/R Designer?**

- You can open the O/R Designer by adding a new **LINQ to SQL Classes** item to a project.

**18. What is the DataContext class and how is it related to LINQ?**

- After you add a **LINQ to SQL Classes** item to a project and open the O/R Designer, the empty design surface represents an empty **DataContext** class ready to be configured. The **DataContext** class is a LINQ to SQL class that acts as a conduit between a SQL Server database and the LINQ to SQL entity classes mapped to that database. This class contains the connection string information and the methods for connecting to a database and manipulating the data in the database. It is configured with connection information provided by the first item that is dragged onto the design surface.

**19. Before you query a DataSet object by using LINQ to DataSet, you must first populate the dataset. How can you do this?**

- You can load the data into the dataset by using different methods, such as:
  - Using the **DataAdapter** class
  - Using LINQ to SQL

**20. What are the different Visual Basic features that support LINQ?**

- Visual Basic includes the following features that support LINQ:
  - **Anonymous types**—Enables you to create a new type based on a query result
  - **Implicitly typed variables**—Enables the compiler to infer and assign a type when you declare and initialize a variable
  - **Extension method**—Enables you to extend an existing type with your own methods without modifying the type itself

**21. The standard query operators are themselves a set of extension methods that provide the LINQ query functionality for any type that implements the **IEnumerable<T>** interface in Visual Basic. Is it True or False?**

- False, as it implements the **IEnumerable<T>** interface in Visual Basic and the **IEnumerable<T>** interface is implemented in C#.

**22. What are lambda expressions in LINQ?**

- A lambda expression is a function without a name that calculates and returns a single value. All lambda expressions use the lambda operator =>, which read as goes to. The left side of the lambda

operator specifies the input parameters and the right side holds the expression or statement block.

**23. In LINQ, lambda expressions underlie many of the standard query operators. Is it True or False?**

- It is true.

**24. Name the control that exposes the LINQ features to Web developers through the ASP.NET data-source control architecture.**

- The LinqDataSource control exposes the LINQ features to Web developers through the ASP.NET data-source control architecture.

**25. What is a LinqDataSource control?**

- The **LinqDataSource** control enables you to use LINQ in an ASP.NET Web page by setting the properties in the markup text. You can use the control retrieve or modify data. It is similar to the **SqlDataSource** and **ObjectDataSource** controls in the sense that it can be used to declaratively bind other ASP.NET controls on a page to a data source. The difference is that instead of binding directly to a database or to a generic class, the **LinqDataSource** control is designed to bind a LINQ enabled data model.

**26. Which command-line tool generates code and mapping for the LINQ to SQL component of .NET Framework?**

- The SqLMetal.exe command-line tool generates code and map the LINQ to SQL component.

**27. Write the name of the property that you need to connect the LinqDataSource control to a database class.**

- Set the **ContextTypeName** property to the name of the class that represents the database. For example, to connect to the **Customers** table in the **northwind** database, set the **ContextTypeName** property to **northwindDataContext** (or whatever name that you specify for the database object).

**28. Which is the new operator added in LINQ?**

- Set operator
- Join operator
- Aggregate operator

- Sorting operator
- Zip operator
- The Zip operator is added in LINQ in .NET Framework 4.0.

**29. Write a note on the Zip operator in .NET Framework 4.0.**

- The Zip operator is a new operator introduced in LINQ in .NET Framework 4.0. It combines two collections from two different sources into one. It uses the Zip() method to merge two sequences, which are identical in length, into one. The Zip() method uses the order of elements to combine the elements from two given sequences. This implies that the Zip() method takes an element from the first as well as from the second source and returns a combined type.

The syntax of the Zip() method is as follows:

In C#:

```
public static TResult IEnumerable<TResult> Zip(
    IEnumerable<TFirst> first,
    IEnumerable<TSecond> second,
    Func<TFirst, TSecond, TResult> selector)
```

In VB:

```
Public Shared Function Zip(first As IEnumerable(Of TFirst),
                           second As IEnumerable(Of TSecond),
                           selector As Func(Of TFirst, TSecond, TResult)) As IEnumerable(Of TResult)
```

**30. What is PLINQ?**

- PLINQ stands for Parallel Language Integrated Query. It is the parallel implementation of LINQ, in which a query can be executed by using multiple processors. PLINQ ensures the scalability of software on parallel processors in the execution environment. It is used where data grows rapidly, such as in telecom industry or where data is heterogeneous. PLINQ also supports all the operators of LINQ. In addition, you can query collections by using PLINQ. It can also run several LINQ queries simultaneously and makes use of the processors

on the system. Apart from this, PLINQ uses parallel execution, which helps in running the queries quickly. Parallel execution provides a major performance improvement to PLINQ over certain types of legacy code, which takes too much time to execute.

### 31. Which extension method do you need to run a parallel query in PLINQ?

- The AsParallel extension method is required to run a parallel query in PLINQ.

# 7

## Dynamic Programming

### 1. What is Dynamic Language Runtime (DLR)?

- DLR is a runtime environment that allows you to integrate dynamic languages with the Common Language Runtime (CLR) by adding a set of services, such as expression trees, call site caching, and dynamic object interoperability to the CLR. The System.Dynamic and System.Runtime.CompilerServices namespaces are used to hold the classes for DLR. It also provides dynamic features to statically-typed languages, such as C# and Visual Basic to enable their interoperation with dynamic languages.

### 2. What are the advantages of DLR?

- The various advantages provided by DLR are:
  - Allows you to easily implement the dynamic languages to the .NET Framework.
  - Provides dynamic features to statically-typed languages. The statically-typed .NET Framework languages, such as C# and Visual Basic can create dynamic objects and use them together with statically-typed objects.
  - Implements sharing of libraries and objects, which means that the objects and libraries implemented in one language can be used by other languages using DLR. The DLR also enables interoperation between statically-typed and dynamic languages.
  - Enables fast execution of dynamic operations by supporting advance caching.

### 3. Give a brief introduction to Binders.

- Binders are used by DLR to communicate with not the .NET Framework but also with various other services, such as Silverlight and COM. These services represent language-specific semantics and specify how a particular operation can be performed at the call site.

Call sites refer to the area in the code where logical and mathematical operations, such as **a + b** or **a.b()** are performed on dynamic objects.

#### 4. Explain the different services provided by DLR to CLR.

- The services provided by DLR to CLR are used for supporting dynamic languages. These services include the following:
  - **Expression Trees**—Refers to the representation of code in a data structure similar to a tree. However, expression trees in DLR are the advanced version of the expression trees that were introduced with LINQ in .NET 3.5. Therefore, DLR has extended the functionalities of Language Integrated Query (LINQ) expression trees, such as control flow, assignment, and other language-modelling nodes to a dynamic language. These expression trees define the semantics of a language in form of an **abstract syntax tree (AST)**. AST enables the DLR to dynamically generate code, which the CLR executes at runtime.

□ **Call Site Caching**—Enables the DLR to store the information of the operations and characteristics of the variables, such as their data type. The call site caching services also enables to check whether such operations have been performed previously to retrieve all the information about the variable. The place where DLR stores these values is called a **call site**.

□ **Dynamic Object Interoperability**—Enables the DLR to provide a set of classes and interfaces that represent dynamic objects and operations. These classes and interfaces can be used to create classes for dynamic libraries, which can be used in static and dynamic type languages.

#### 5. Name the binders provided by .NET Framework 4.0.

- .NET Framework 4.0 provides the following binders:
  - **Object Binder**—Enables to communicate with .NET objects
  - **JavaScript Binder**—Enables to communicate with JavaScript in Silverlight
  - **Python Binder**—Enables to communicate with IronPython
  - **Ruby Binder**—Enables to communicate with IronRuby
  - **COM Binder**—Enables to communicate with COM

#### 6. What is Dynamic Data Type?

- Dynamic data type is a reference data type, which is checked at the run time instead of the compile time; thereby, allowing you to detect the errors at run time only. The following example shows the declaration of the dynamic data type:

```
// In this case, 'floatDyn' and 'stringDyn'
// are the name of variables whose data types
// are dynamic.

dynamic floatDyn = 4.46;
Console.WriteLine(floatDyn);
dynamic stringDyn = "EnhancedType";
Console.WriteLine(stringDyn);
```

#### 7. What is the difference between dynamic and var data types?

- The difference between the var and dynamic data types is that the var data type is strongly type checked at the compile time; whereas, the dynamic data type is type checked by the compiler only at run time. After declaring a var data type, you cannot explicitly change its type throughout the execution of the program; however, a variable of the dynamic data type can be changed during runtime. Another major difference between the two is that dynamic type can also be used as the return type for methods, for which var cannot be used.

#### 8. Which class is used for converting the data types?

- The System.Convert class provides a complete set of methods for converting the data types.

#### 9. Explain ExpandoObject and DynamicObject classes.

- The ExpandoObject class refers to a class whose members can be explicitly added and removed at runtime. In other words, the ExpandoObject class allows dynamic binding of the objects, which enables you to use standard syntax, similar to the **dynobj.Method** method instead of using more complex syntax, such as **dynobj.getAttribute("Method")**.

The `DynamicObject` class enables you to define the dynamic behavior for an object at run time. This class cannot be instantiated directly; therefore, to implement the dynamic behavior, you must inherit from the `DynamicObject` class and override the necessary methods. It allows you to define the specific operations that can be performed on dynamic objects as well the methods to perform those operations.



## Extensible Markup Language (XML)

### 1. Write a note on Extensible Markup Language (XML).

- XML is a simple and flexible markup language in the text format. Nowadays, it is widely used to exchange a large variety of data over the Internet. XML consists of data as text in well-defined customized layouts by using self-defining tags. These user-defined tags are user friendly because they contain the name given by the user and make the information easily understandable to a user. These user-friendly features made XML to be widely used as a standard data-interchange format. The World Wide Web Consortium (W3C) frequently develops new standard for XML usage by different software vendors and solution providers. XML plays a very significant role with respect to .NET Framework 4.0. .NET Framework 4.0 provides us with a namespace called `System.Xml`, which includes classes that are used to work with XML.

### 2. Explain the difference between XML and HTML.

- XML and HTML are markup languages with similar features, but still both have many diversities, such as:
  - XML focuses on describing and carrying data. It tries to display data in a more understandable manner while HTML focuses on data outlooks. It tries to present data in more presentable manner.
  - XML supports user-defined tags while HTML provides pre-defined tags.
  - XML is a case-sensitive language while HTML language is not case-sensitive.
  - In XML, all tags must be closed; while in HTML, it is not necessary to close each tag.

### 3. Describe the rules and regulations that must be followed while creating a well-formed XML document.

- The following are the rules and regulations that are necessary to follow while creating a well-formed XML document:

- Every start tag must end with an end tag.
- A root element should be included for enclosing other child elements.
- XML tags are case-sensitive; therefore, start and end tags must be of same spelling and the casing should also be the same.
- XML's empty tags are necessary to close with a forward slash (/).
- XML's attributes values are necessary to enclose within double quotation marks.
- XML tags must be properly nested. It means starting tags should be closed in the reverse order in which they present.

#### 4. What is an XML attribute?

- An XML attribute contains additional information regarding that particular element. The XML attributes use the name-value pair. For example, the element Student has an attribute called ID and the value of this attribute is set to s01, as shown in the following code snippet:

```
<Student ID="s01">
</Student>
```

#### 5. Explain the XML elements.

- The elements are the central units of an XML document that explain and identify data. The elements are represented by the tags. You can also make your own tags, which make XML a user-friendly language. By creating custom meaningful elements, you can improve readability of the document. XML elements can be nested and the nested elements are known as child elements.

#### 6. What are the naming conventions required for XML elements tags?

- The following are the naming conventions that need to be followed for XML elements tags:
- Element names should contain only characters, numbers, hyphens, and periods
  - Element names cannot not begin with a number or punctuation character
  - Element names must not start with the word xml (or XML, or Xml)
  - Element names cannot consist spaces
  - Element names can be used any words except xml, XML, or Xml because no words are reserved in XML

#### 7. What is the DTD?

- The DTD is Document Type Definition that describes the formation of the content of an XML document. The DTD manages the data to store in a consistent format. It defines the XML elements and attributes about how they should be present in XML documents and what relation they should have with other elements and attributes. The DTD also allows you to mention whether an XML element is optional or not. If the XML documents are not according to the DTD rules, they are not considered valid.

#### 8. What is an XML schema?

- An XML Schema provides the definition of an XML document. This implies that an XML schema defines the following in an XML document:
- The elements that can appear in an XML document
  - The attributes that can appear in an XML document
  - The elements that are child elements
  - The order of child elements
  - The number of child elements
  - Whether an element is empty or it includes some text
  - The data types for elements and attributes

#### 9. State the advantages of XML schemas over DTD.

- Microsoft developed a language known as the XML Schema Definition (XSD) to describe the schema to an XML document. The following are the advantages of XML schemas over DTDs:
- XSD keeps much better control over types of data than the DTD.
  - DTD does not allow creating customized data types while the XSD provides full support to create customized data types.
  - XSD allows you to specify restrictions on data. It means that you can define the type of data that should be stored in an element, for example numbers or alphabets.
  - The XSD is quite easy to learn and to understand because its syntax is same as that of the XML document.

#### 10. What is an XML namespace?

- The XML namespace allows you to avoid clashes to locate the same name of elements that are presented in more than one XML document. For example, <student id = "1001">...</student> can occur in the College.xml and Library.xml files. If a program merges these two files, then a conflict arises to accurately

locate the <student> element. Therefore, XML namespaces are declared by using the namespace attribute of the <xmlns> element. The syntax for declaring a XML namespace is given as follows:

`<xm:ns:namespace prefix="namespaceuri">`

With the help of namespace, you can easily refer to a collection of elements and their attributes. When you assign an XML parent element to a namespace automatically, all child elements and attributes of that element are assigned in that namespace. The parent element uses the prefixes of namespaces to attach its child elements and attributes in that particular namespace.

The following is an example to declare a single namespace prefix for multiple elements:

```
<air:aircraft xmlns:air="http://www.aircrafts.com">
  <air:names>MIG</air:names>
  <air:number>2309</air:number>
</air:aircraft>
```

### 11. Which namespaces in .NET are used for XML?

- The System.xml.dll is the real physical file, which contains the XML implementation. Some of the other namespaces that allow .NET to use XML are as follows:

- System.Xml
- System.Xml.Schema
- System.Xml.XPath
- System.Xml.Xsl

### 12. The XML preserves white spaces. Is it true?

- Yes, it is true.

### 13. The XML elements cannot be empty. Is it true?

- No, it is not true.

### 14. Is it true that the XML's goal is to replace HTML?

- No, it is not true. Both are necessary in their respective fields.

### 15. Out of the following four options, which one is the correct syntax to define the XML version in a XML document?

- a. <?xml version="1.0" />
  - b. <?xml version="1.0"?>
  - c. <xml version="1.0" />
  - d. <?xml version="1.0">
- The correct answer is b.

### 16. Is the following document a well-formed XML document?

```
<?xml version="1.0"?>
<student>
  <id>C001</id>
  <name>Tom</name>
  <contactno>4545655</contactno>
</students>
```

- No, it is not a well-formed XML document because the root element is missing in it.

### 17. Is the following document a well-formed XML document?

```
<?xml Version="1.0"?>
<student>
  <student age=13>James</student>
  <contactno>4523678</contactno>
</students>
```

- No, this is not a well-formed XML document because the attribute value is not enclosed within double quotation marks.

### 18. Which of the following is a correct name for an XML element?

- a. <1dollar>
  - b. <xmldocument>
  - c. <first name>
  - d. <Name5>
- The correct answer is d.

### 19. Which is the correct way of referring to a style sheet (styles.xsl)?

- a. <?xmlstylesheet type="text/xsl" href="styles.xsl" ?>
  - b. <link type="text/xsl" href="styles.xsl" />
  - c. <stylesheet type="text/xsl" href="styles.xsl" />
- The correct answer is a.

### 20. Which are the five special characters that cannot be included in an XML document?

- Table 8.1 lists the five special characters that cannot be included in an XML document and the predefined entities that have replaced them.

Table 8.1 Showing Special Character with the Predefined Entity:

Special Character	Predefined Entity Reference
<	&lt;
>	&gt;
&	&amp;

Special character	Predefined Entity Reference
"	&quot;
<	&lt;

**21. What is XSLT?**

- XSLT is Extensible Stylesheet Language Transformations that is a part of XML, which is a mechanism to transform an XML document into another XML or HTML document.

**22. Using XSLT, how would you extract the value of a specific attribute from an element in an XML document?**

- The components necessary for the above mentioned operation are as follows:
  - The template element—Matches the correct XML element
  - The value-of element—Selects the attribute value
  - The optional apply-templates element—Allows continuous processing of the document

**23. Describe the role that XSL can play while dynamically generating HTML pages from a relational database.**

- The SQLXML 3.0 and advanced versions provide the facility of mapping the SQL queries output with XSLT templates. It uses XSLT to present the records that are retrieved from databases on Web pages (HTML pages).

An application can use XSLT to modify the output that is retrieved from data sources and display the output by XSLT templates. The XSLT displays data without affecting the database query and the code of application.

**24. What is XPath?**

- XPath stands for XML Path. It is a language used to access different parts of an XML document, such as elements and attributes.

**25. What is XML DOM?**

- The DOM stands for Document Object Model, which describes the logical formation of documents and provides the way to access and manipulate a document. It supplies an Application Programming Interface (API) to XML documents. It is built around the object-oriented design; therefore, it is known as DOM. The DOM model considers an XML document as a composition of objects and every object consists of properties and behaviors that can be manipulated.

**26. Give an example of a DOM-enabled XML parser.**

- The XML parser is MSXML, which is fully DOM-enabled.

**27. Explain different types of XML Application Programming Interface (API).**

- The following are two main types of XML parsers:
  - Tree-based API —Compiles an XML document into a tree structure and loads it into memory. You can traverse and change the tree structure. The DOM is an example of a tree-based API.
  - Event-based API —Provides the report to an application about the parsing events by a set of built-in callback functions. An example of the event-based API is SAX.

**28. What are the advantages of DOM?**

- The following are the advantages of DOM:
  - DOM stores the entire XML document into memory before processing. Therefore, the XML structure can be easily modified and values can be added, changed, and removed.
  - DOM enables to traverse the XML structure in any direction. It means that you can access any node of the XML structure by traversing through the XML structure.

**29. Which classes are supported to make an XML DOM?**

- The following are the different classes in the System.Xml namespace that make up the XML DOM:
  - The XmlNode class
  - The XmlDocument class
  - TheXmlElement class
  - TheXmlAttribute class
  - The XmlText class
  - The XmlComment class
  - The XmlNodeList class

**30. Which class is used to encode and decode XML names and contains different methods to convert between CLR types and XSD types.**

- The `XmlConvert` class.

**31. Name the W3C standards supported by .NET Framework.**

■ There are many standards set by W3C, which are supported by .NET Framework 4.0:

- XML 1.0 ([www.w3.org/TR/1998/REC-xml1-19980210](http://www.w3.org/TR/1998/REC-xml1-19980210)) with Document Type Definition (DTD) support
- XML Namespaces ([www.w3.org/TR/REC-xml-names](http://www.w3.org/TR/REC-xml-names)) for both stream-level and DOM
- XML Schemas ([www.w3.org/2001/XMLSchema](http://www.w3.org/2001/XMLSchema))
- XPath expressions ([www.w3.org/TR/xpath](http://www.w3.org/TR/xpath))
- XSLT transformations ([www.w3.org/TR/xslt](http://www.w3.org/TR/xslt))
- DOM Level 1 Core ([www.w3.org/TR/REC-DOM-Level-1/](http://www.w3.org/TR/REC-DOM-Level-1/))
- DOM Level 2 Core ([www.w3.org/TR/REC-DOM-Level-2-Core/](http://www.w3.org/TR/REC-DOM-Level-2-Core/))
- SOAP 1.1 ([www.w3.org/TR/SOAP](http://www.w3.org/TR/SOAP))

**32. Explain the `XmlReader` class.**

■ The `XmlReader` class is used to read XML data in a fast, forward-only, and non-cached manner.

To work with `XmlReader` class in .NET, you need to import the following namespace:

*In C#:*

`using System.Xml;`

*In VB:*

`<Imports System.Xml>`

**33. Describe the `XmlWriter` class.**

■ The `XmlWriter` class is used to write XML to a stream, a file, or a `TextWriter` object. This class works in a forward-only, non-cached manner. You can configure the `XmlWriter` object up to a large extent. With this object, you can specify a few things, such as whether to indent content or not, the amount to indent, what quote character to use in attribute values, and whether or not namespaces are supported.

**34. What is the function of the  `XmlDocument` class?**

■ The  `XmlDocument` class represents the entire XML document and it implements the DOM Level 1 and DOM Level 2 specifications. Unlike the  `XmlReader` and  `XmlWriter` classes, this class gives you read and write capabilities and random access to the DOM tree.

# 9

## ASP.NET

**1. What is ASP?**

■ Active Server Pages (ASP), also known as Classic ASP, is a Microsoft's server-side technology, which helps in creating dynamic and user-friendly Web pages. It uses different scripting languages to create dynamic Web pages, which can be run on any type of browser. The Web pages are built by using either VBScript or JavaScript and these Web pages have access to the same services as Windows application, including ADO (ActiveX Data Objects) for database access, SMTP (Simple Mail Transfer Protocol) for e-mail, and the entire COM (Component Object Model) structure used in the Windows environment. ASP is implemented through a dynamic-link library (`asp.dll`) that is called by the IIS server when a Web page is requested from the server.

**2. What is ASP.NET?**

■ ASP.NET is a specification developed by Microsoft to create dynamic Web applications, Web sites, and Web services. It is a part of .NET Framework. You can create ASP.NET applications in most of the .NET compatible languages, such as Visual Basic, C#, and J#. The ASP.NET compiles the Web pages and provides much better performance than scripting languages, such as VBScript. The Web Forms support to create powerful forms-based Web pages. You can use ASP.NET Web server controls to create interactive Web applications. With the help of Web server controls, you can easily create a Web application.

**3. 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).

#### 4. What is an ASP.NET Web Form?

- ASP.NET Web forms are designed to use controls and features that are almost as powerful as the ones used with Windows forms, and so they are called as Web forms. The Web form uses a server-side object model that allows you to create functional controls, which are executed on the server and are rendered as HTML on the client. The attribute, runat="server", associated with a server control indicates that the Web form must be processed on the server.

#### 5. Explain the new features of ASP.NET 4.0 in brief.

- The following are the new features of ASP.NET 4.0:
  - The ASP.NET 4.0 core services include features, such as output caching and session-state storage. The benefits of the ASP.NET core services are as follows:
    - The major configuration elements of the web.config file have been moved to the machine.config file. This implies that the web.config file in this new version of ASP.NET is either empty or consists of some of the basic code, which specifies the version of the Visual Studio framework that the application is targeting.
    - The output caching feature in ASP.NET enables the developers to store the generated output of controls and HTTP responses in the memory. This is very helpful as in case of subsequent Web requests, the content can be retrieved from the output.
    - The new version of ASP.NET provides a better approach for starting an application pool, initializing an ASP.NET application, and accepting HTTP requests.
    - ASP.NET 4.0 removes the unnecessary round trip made by the browser for temporary redirects.
    - ASP.NET 4.0 reduces the size of session-state data.
    - In this version of ASP.NET, there is an increase in the size of application URLs, by using the new httpRuntime configuration attribute.
  - The Web forms form an integral part of ASP.NET applications. It has been enhanced several times since its release. In ASP.NET 4.0, the following new features are introduced in Web forms:
    - Setting meta tags by using the MetaKeywords and MetaDescription properties of the Page class
    - Disabling view state for all controls or enabling view state for individual controls on a page by using the ViewStateMode property of the Control class

(Model) and user interface (View) are separated from each other so that any change in the user interfaces does not affect the data in the database and vice versa.

- Cascading Style Sheet (CSS) is one of the important components of a Web application, as it makes the designing of a Web site easier and faster. The Web site in Visual Studio 2010 has been updated to include the improved CSS 2.1 standards. These standards are more robust than previous versions of ASP.NET and help in preserving the integrity of Web applications and their HTML code.

#### 6. What is IIS? Why is it used?

■ Internet Information Services (IIS) is created by Microsoft to provide Internet-based services to ASP.NET Web applications. It makes your computer to work as a Web server and provides the functionality to develop and deploy Web applications on the server. IIS handles the request and response cycle on the Web server. It also offers the services of SMTP and FrontPage server extensions. The SMTP is used to send emails and use FrontPage server extensions to get the dynamic features of IIS, such as form handler.

#### 7. Among the following, which is the latest version of IIS?

- 5.0
- 5.1
- 8.0
- 7.0
- 7.5

- The latest version of IIS is 7.5.

#### 8. What are the events that happen when a client requests an ASP.NET page from IIS server?

- The following events happen when a client requests an ASP.NET page from the IIS server:

- User requests for an application resource
- The integrated request-processing pipeline receives the first user request
- Response objects are created for each user request
- An object of the `HttpApplication` class is created and allocated to the Request object
- The `HttpApplication` class processes the user request

#### 9. How would ASP and ASP.NET applications run at the same time on the same server?

Or

#### How does IIS handle an ASP and ASP.NET request?

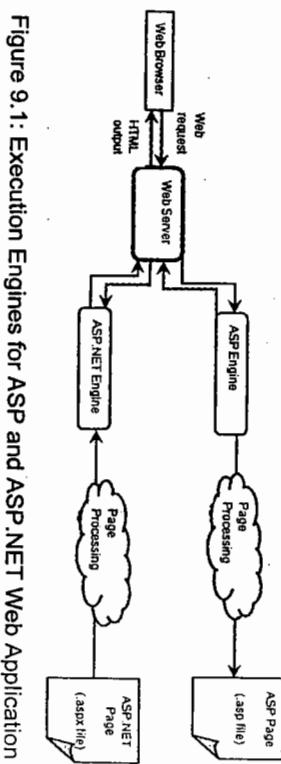


Figure 9.1: Execution Engines for ASP and ASP.NET Web Application

#### 10. What is the process of the server-side application architecture?

■ In the server-side Web application, when the client makes a request for a Web page, the request is sent to the server. The server after getting the request runs the server-side application and returns back the output as a response in the form of an HTML document. The whole process of the server-side Web application architecture is shown in Figure 9.2:

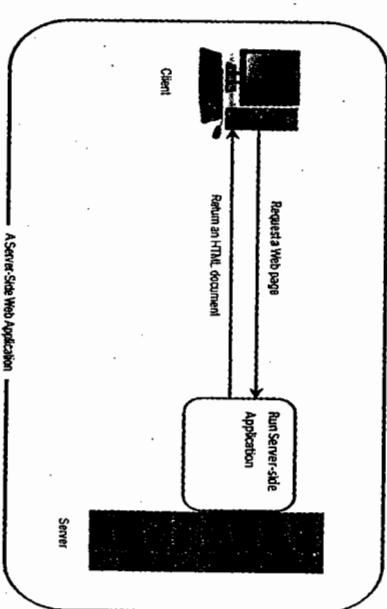


Figure 9.2: Server-Side Web Application

#### 11. What is the process of the client-side application architecture?

■ In the client-side Web application architecture, when a user makes a request, the request is sent to the server. The server processes the request and sends back the response to the client. The client then runs the client-side application with the response provided by the

server. The whole process of the client-side Web application architecture is shown in Figure 9.3:

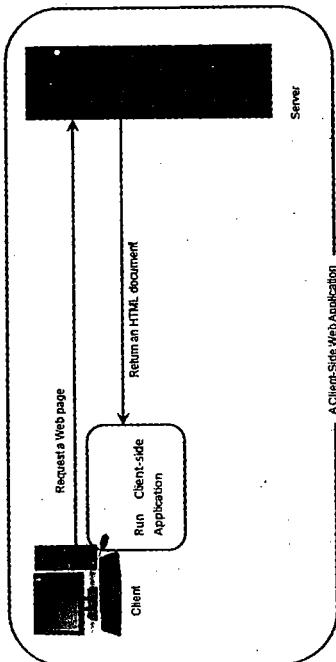


Figure 9.3: Client-Side Web Application

12. What is actually returned from server to the browser when a browser requests an .aspx file and the file is displayed?
- When a browser requests an .aspx file then the server returns a response, which is rendered into a HTML string.

13. What events are fired when a page loads?

- The following events fire when a page loads:
  - Init()*—Fires when the page is initializing
  - LoadViewState()*—Fires when the view state is loading
  - LoadpostData()*—Fires when the postback data is processing
  - Load()*—Fires when the page is loading
  - PreRender()*—Fires at the brief moment before the page is displayed to the user as HTML
  - Unload()*—Fires when the page is destroying the instances of server controls

14. Which event determines that all the controls are completely loaded into memory?

- The *Page\_Load* event determines that all the controls on the page are fully loaded. You can also access the controls in the *Page\_Init* event; however, the *ViewState* property does not load completely during this event.

15. To which class a Web form belongs to in the .NET Framework class hierarchy?

- A Web form belongs to the `System.Web.UI.Page` class.

16. How information about the user's locale can be accessed?
- The information regarding a user's locale can be accessed by using the `System.Web.UI.Page.Culture` property.

17. What is a round trip?

- The trip of a Web page from the client to the server and then back to the client is known as a round trip.

18. How can you identify that the page is PostBack?

- The `Page` object uses the `IsPostBack` property to check whether the page is posted back or not. If the page is postback, this property is set to true.

19. What are the HTML server controls in ASP.NET?

- HTML server controls are similar to the standard HTML elements, which are normally used in HTML pages. They expose properties and events that can be used programmatically. To make these controls programmatically accessible, you need to specify that the HTML controls act as a server control by adding the `runat="server"` attribute.

20. What are Web server controls in ASP.NET?

- The ASP.NET Web server controls are objects on the ASP.NET pages that run when the Web page is requested. Many Web server controls, such as button and text box, are similar to the HTML controls. In addition to the HTML controls, there are many controls, which include complex behavior, such as the controls used to connect to data sources and display data.

21. What are Custom User Controls in ASP.NET?

- The custom user controls are the controls that are defined by developers. These controls are a mixture of custom behavior and predefined behavior. These controls work similar to other Web server controls.
  -

22. What is the difference between Web User Control and Web Custom Control in ASP.NET?

- You can find the differences between Web User Control and Web Custom Control in ASP.NET in Table 9.1.

Table 9.1: Difference between Web User Control and Web Custom Control:

Web User Control	WebCustomControl
Easy to create	Difficult to create
Limited support for users who use visual design tool	Full visual design tool support for users
A separate copy of the control is required in each application	A single copy of the controls compiled assembly is registered into the global assembly cache (GAC) and later on used in different Web applications

If cannot be added to the toolbox in Visual Studio IDE	It can be added to the Toolbox in Visual Studio IDE
Good for static layout	Good for dynamic layout

### 23. What are the major built-in objects in ASP.NET?

The major built-in objects in ASP.NET are as follows:

- Application
- Response
- Server
- Session
- Context

### 24. What are HTTP handlers in ASP.NET?

- HTTP handlers, as the name suggests, are used to handle user requests for Web application resources. They are the backbone of the request-response model of Web applications. There is a specific event handler to handle the request for each user request type and send back the corresponding response object.

Each user request to the IIS Web server flows through the HTTP pipeline, which refers to a series of components (HTTP modules and HTTP handlers) to process the request. HTTP modules act as filters to process the request as it passes through the HTTP pipeline. The request, after passing through the HTTP modules, is assigned to an HTTP handler that determines the response of the server to the user request. The response then passes through the HTTP modules once again and is then sent back to the user.

You can define HTTP handlers in the <httpHandlers> element of a configuration file. The <add> element tag is used to add new handlers and the <remove> element tag is used to remove existing

handlers. To create an HTTP handler, you need to define a class that implements the IHttpHandler interface.
---

### 25. Explain the concept of states in ASP.NET.

- State is quite an innovative concept in Web development because it eliminates the drawback of losing state data due to reloading of a Web page. By using states in a Web application, you can preserve the state of the application either at the server or client end. The state of a Web application helps you to store the runtime changes that have been made to the Web application. For example, as already described earlier, a change in the data source of the Web application products in the shopping cart. If you are not using states, these changes are discarded and are not saved. You may think that the whole concept of storing states is optional. However, under certain circumstances, using states with applications is imperative. For example, it is necessary to store states for Web applications, such as an e-commerce shopping site or an Intranet site of a company, to keep track of the requests of the users for the items they have selected on the shopping site or the days requested for vacation on the Intranet site.

### 26. Explain the Application and Session objects in ASP.NET.

- Application state is used to store data corresponding to all the variables of an ASP.NET Web application. The data in an application state is stored once and read several times. Application state uses the HttpApplicationState class to store and share the data throughout the application. You can access the information stored in an application state by using the HttpSession class property. Data stored in the application state is accessible to all the pages of the application and is the same for all the users accessing the application. The HttpSession class provides a lock method, which you can use to ensure that only one user is able to access and modify the data of an application at any instant of time.
- Each client accessing a Web application maintains a distinct session with the Web server, and there is also some specific information associated with each of these sessions. Session state is defined in the <sessionState> element of the web.config file. It also stores the data specific to a user session in session variables. Different session variables are created for each user session. In addition, session variables can be accessed from any page of the application. When a user accesses a page, a session ID for the user is created. The session ID is transferred between the server and the client over the HTTP protocol using cookies.

**27. What is State Management? How many ways are there to maintain a state in .NET?**

- State management is used to store information requests. The state management is used to trace the information or data that affect the state of the applications.

There are two ways to maintain a state in .NET, Client-Based state management and Server-Based state management.

The following techniques can be used to implement the Client-Based state management:

- View State
- Hidden Fields
- Cookies
- Query Strings
- Control State
- Session State
- Profile Properties

**28. What is the code-behind feature in ASP.NET?**

- The code-behind feature of ASP.NET enables you to divide an ASP.NET page into two files—one consisting of the presentation data, and the second, which is also called the code-behind file, consisting of all the business logic. The presentation data contains the interface elements, such as HTML controls and Web server controls, and the code-behind contains the event-handling process to handle the events that are fired by these controls. The file that contains the presentation data has the .aspx extension. The code behind file has either the .cs extension (if you are using the programming language C#) or the .vb (if you are using the programming language Visual Basic .NET) extension.

**29. What are the advantages of the code-behind feature?**

- The code-behind feature of ASP.NET offers a number of advantages:
  - Makes code easy to understand and debug by separating application logic from HTML tags
  - Provides the isolation of effort between graphic designers and software engineers
  - Removes the problems of browser incompatibility by providing code files to exist on the Web server and supporting Web pages to be compiled on demand

**30. Suppose you want an ASP.NET function (client side) executed on the MouseOver event of a button. Where do you add an event handler?**

- The event handler is added to the Add method of the Attributes property.

**31. Which ASP.NET objects encapsulate the state of the client and the browser?**

- The Session object encapsulates the state of the client and browser.

**32. What is the behavior of a Web browser when it receives an invalid element?**

- The behavior of a Web browser when it receives an invalid element depends on the browser that you use to browse your application. Most of the browsers ignore the invalid element; whereas, some of them display the invalid elements on the page.

**33. Which class is inherited when an ASP.NET server control is added to a Web form?**

- The System.Web.UI.WebControls class is inherited when an ASP.NET server control is added to a Web form.

**34. Explain the validation controls. How many validation controls in ASP.NET 4.0?**

- Validation controls are responsible to validate the data of an input control. Whenever you provide any input to an application, it performs the validation and displays an error message to user, in case the validation fails.

ASP.NET 4.0 contains the following six types of validation controls:

- CompareValidator—Performs a comparison between the values contained in two controls
- CustomValidator—Writes your own method to perform extra validation
- RangeValidator—Checks value according to the range of value
- RegularExpressionValidator—Ensures that input is according to the specified pattern or not
- RequiredFieldValidator—Checks either a control is empty or not
- ValidationSummary—Displays a summary of all validation error in a central location

**35. How can you display all validation messages in one control?**

- The ValidationSummary control displays all validation messages in one control.

**36. Which data type does the RangeValidator control support?**

- The data types supported by the RangeValidator control are Integer, Double, String, Currency, and Date.

**37. Which control will you use to ensure that the values in two different controls match?**

- You should use the CompareValidator control to ensure that the values in two different controls match.

**38. Can you post and access view state in another application?**

- Yes, you can post and access a view state in other applications. However, while posting a view state in another application, the PreviousPage property returns null.

**39. What is the function of the CustomValidator control?**

- It provides the customize validation code to perform both client-side and server-side validation.

**40. What happens if an ASP.NET server control with event-handling routines is missing from its definition?**

- The compilation of the application fails.

**41. Which method is used to force all the validation controls to run?**

- The Page.Validate() method is used to force all the validation controls to run and to perform validation.

**42. Where is the ViewState information stored?**

- The ViewState information is stored in the HTML hidden fields.

**43. What does the "EnableViewState" property do? Why do we want it On or Off?**

- The EnableViewState property enables the ViewState property on the page. It is set to On to allow the page to save the users input between postback requests of a Web page; that is, between the Request and corresponding Response objects. When this property is set to Off, the page does not store the users input during postback.

**44. What is the use of web.config? What is the difference between machine.config and web.config?**

- ASP.NET configuration files are XML-based text files for application-level settings and are saved with the name web.config. These files

are present in multiple directories on an ASP.NET Web application server. The web.config file sets the configuration settings to the directory it is placed in and to all the virtual sub folders under it. The settings in sub directories can optionally override or change the settings specified in the base directory. The difference between the web.config and machine.config files is given as follows:

- <WinDir>\Microsoft\NET\Framework\<version>\config\machine.config provides default configuration settings for the entire machine. ASP.NET configures IIS to prohibit the browser directly from accessing the web.config files to make sure that their values cannot be public. Attempts to access those files cause ASP.NET to return the 403: Access Forbidden error.
- ASP.NET uses these web.config configuration files at runtime to compute hierarchically a sole collection of settings for every URL target request. These settings compute only once and cached across further requests. ASP.NET automatically checks for changing file settings and do not validate the cache if any of the configuration changes made.

**45. What are the various ways of authentication techniques in ASP.NET?**

- There are various techniques in ASP.NET to authenticate a user. You can use one of the following ways of authentication to select a built-in authentication provider:
  - *Windows Authentication* —This mode works as the default authentication technique. It can work with any form of Microsoft Internet Information Services (IIS) authentication, such as Basic, Integrated Windows authentication (NTLM/Kerberos), Digest, and certificates. The syntax of Windows authentication mode is given as follows:
 

```
<authentication mode="Windows"/>
```
  - *Forms Authentication* —You can specify this mode as a default authentication mode by using the following code snippet:
 

```
<authentication mode="Forms"/>
```
  - *Passport* —This mode works with Microsoft Passport authentication, as shown in the following code snippet:
 

```
<authentication mode="Passport"/>
```

**46. What is Role-based security?**

- In the Role-based security, you can assign a role to every user and grant the privilege according to that role. A role is a group of principal that restricts a user's privileges. Therefore, all the organization and applications use role-based security model to

determine whether a user has enough privileges to perform a requested task.

#### 4.7. What is the appSettings Section in the web.config file?

- The web.config file sets the configuration for a Web project. The appSettings block in configuration file sets the user-defined values for the whole application.

For example, in the following code snippet, the specified ConnectionString section is used throughout the project for database connection:

```
<appSettings>
  <add key="ConnectionString" value="mydb" />
</appSettings>
```

#### 4.8. What is the use of the <sessionState> tag in the web.config file?

- The <sessionState> tag is used to configure the session state features. To change the default timeout, which is 20 minutes, you have to add the following code snippet to the web.config file of an application:

```
<sessionState timeout="20" />
```

#### 4.9. What is Query String? What are its advantages and limitations?

- The Query String helps in sending the page information to the server.

The Query String has the following advantages:

- Every browser works with Query Strings
- It does not require server resources and so does not exert any kind of burden on the server

The following are the limitations of Query String:

- Information must be within the limit because URL does not support many characters
- Information is clearly visible to the user, which leads to security threats

#### 50. What is a Cookie? Where is it used in ASP.NET?

- Cookie is a lightweight executable program, which the server posts to client machines. Cookies store the identity of a user at the first visit of the Web site and validate them later on the next visits for their authenticity. The values of a cookie can be transferred between the user's request and the server's response.

#### 51. How many types of Cookies are available in ASP.NET?

- There are two types of Cookies available in ASP.NET:
  - Session Cookie —Resides on the client machine for a single session until the user does not log out.
  - Persistent Cookie —Resides on a user's machine for a period specified for its expiry, such as 10 days, one month, and never. The user can set this period manually.

#### 52. What is the default timeout for a Cookie?

- The default time duration for a Cookie is 30 minutes.

#### 53. Explain how Cookies work. Give an example of Cookie abuse.

- The server tells the browser to put some files in a cookie, and the client then sends all the cookies for the domain in each request. An example of cookie abuse is large cookies affecting the network traffic.

#### 54. Which method do you use to kill explicitly a users session?

- The Session.Abandon() method kills the user session explicitly.

#### 55. What are the different ways to send data across pages in ASP.NET?

- The following two ways are used to send data across pages in ASP.NET:
  - Session
  - public properties

#### 56. How do you sign out from forms authentication?

- The FormsAuthentication.Signout() method is used to sign out from the forms authentication.

#### 57. What is the use of the Global.asax file?

- The Global.asax file executes application-level events and sets application-level variables.

#### 58. What are the event handlers that can be included in the Global.asax file?

- The Global.asax file contains some of the following important event handlers:
  - Application\_Error
  - Application\_Start
  - Application\_End

- Session\_Start
- Session\_End

## 59. Which namespaces are necessary to create a localized application?

- The System.Globalization and System.Resources namespaces are essential to develop a localized application.

## 60. What is tracing? Where is it used?

- Tracing displays the details about how the code was executed. It refers to collecting information about the application while it is running. Tracing information can help you to troubleshoot an application. It enables you to record information in various log files about the errors that might occur at run time. You can analyze these log files to find the cause of the errors.

In .NET, we have objects called Trace Listeners. A listener is an object that gets the trace output and stores it to different places, such as a window, a file on your locale drive, or a SQL Server.

The System.Diagnostics namespace contains the predefined interfaces, classes, and structures that are used for tracing. It supplies two classes, Trace and Debug, which allow you to write errors and logs related to the application execution. Trace listeners are objects that collect the output of tracing processes.

## 61. Which ASP.NET configuration options are supported in the ASP.NET implementation on the shared Web hosting platform?

- There are many ASP.NET configuration choices, which are not able to configure at the site, application, or child directory level on the shared hosting environment. Some options can produce security, performance, and stability problem to the server and therefore cannot be changed. The following settings are the only ones that can be changed in the web.config file(s) of your Web site:

- browserCaps
- clientTarget
- pages
- customErrors
- globalization
- authorization
- authentication
- webControls
- webServices

## 62. What are server-side comments?

■ Server-side comments are included in an ASP.NET page for the purpose of documentations as shown in the following code snippet:  
`<%-- This is a sample example of server-side comments --%>`

The server-side comments begin with `<%--` and end with `--%>`.

## 63. How can you dynamically add user controls to a page?

- User controls can be dynamically loaded by adding a Web User Control page in the application and adding the control on this page.

## 64. What is the difference between authentication and authorization?

- Authentication verifies the identity of a user and authorization is a process where you can check whether or not the identity has access rights to the system. In other words, you can say that authentication is a procedure of getting some credentials from the users and verify the user's identity against those credentials. Authorization is a procedure of granting access of particular resources to an authenticated user. You should note that authentication always takes place before authorization.

## 65. Is there a way to set Option Explicit and Option Strict in an ASP.NET page without using a code-behind?

- Yes, you can set Option Explicit and Option Strict in an ASP.NET page without using a code-behind by adding the following code snippet:  
`<%@ Page Language="VB" Explicit="False" Strict="True" %>`

## 66. How can you upload a file from an ASP.NET page?

- The FileUpload control on ASP.NET is used to support the file uploading functionality. This control appears as a combination of the TextBox and the Button control and uploads a file from the client's system to the Web server. Users can click the button to browse in their local drives to select a file.

To upload a file in ASP.NET, you have to use two classes: the `HttpInputFile` class and the `HttpPostedFile` class. The `HttpInputFile` class provides an HTML input type control that the user uses to select a file to upload. The `HttpPostedFile` class provides the uploaded file. To use the `HttpInputFile` control, you must include the `enctype` attribute to your form tag as follows:

```
<form id="upload" method="post" enctype="multipart/form-data">
```

**80. Unlike the Menu or Treeview control, the SiteMapPath control does not use a SiteMapDataSource control. Instead, it relies on a SiteMapProvider to retrieve the data that it will display. Is it true or false?**

- It is True.

**81. Differentiate between client-side and server-side validations in Web pages.**

- Client-side validations take place at the client end with the help of JavaScript and VBScript before the Web page is sent to the server. On the other hand, server-side validations take place at the server end.

**82. Where should the data validations be performed—at the client side or at the server side and why?**

- Data validations should be done primarily at the client side and the server-side validation should be avoided because it makes server task overloaded. If the client-side validation is not available, you can use server-side validation. When a user sends a request to the server, the validation controls are invoked to check the user input one by one.

**83. How do you prevent a validation control from validating data at the client end?**

- You can prohibit a validation control to validate data at the client side by setting the EnableClientScript property to False.

**84. How can you check if all the validation controls on a Web page are valid and proper?**

- You can determine that all the validation controls on a Web page are properly working by writing code in the source file of the Web page using a scripting language, such as VBScript or JavaScript. To do this task, you have to loop across validators collection of pages and check the IsValid property of each validation control on the Web page to check whether or not the validation test is successful.

**85. What does the WebpartListUserControlPath property of a DeclarativeCatalogPart control do?**

- The WebpartListUserControlPath property sets the route of the user defined control to a DeclarativeCatalogPart control.

**86. What does the .WebPart file do?**

- The .WebPart file explains the settings of a Web Parts control that can be included to a specified zone on a Web page.

**87. In which database is the information, such as membership, role management, profile, and Web parts personalization, stored?**

- The aspnetdb database stores all information.

**88. What type of the CatalogPart control enables users to restore the Web Parts that have been removed earlier by the user?**

- The PageCatalogPart control.

**89. How does a content page differ from a master page?**

- A content page does not have complete HTML source code; whereas a master page has complete HTML source code inside its source file.

**90. Why do we need nested master pages in a Web site?**

- When we have several hierarchical levels in a Web site, then we use nested master pages in the Web site.

**91. How will you differentiate a submaster page from a top-level master page?**

- Similar to a content page, a submaster page also does not have complete HTML source code; whereas, a top-level master page has complete HTML source code inside its source file.

**92. Where is a skin file stored in a web application?**

- a. App\_Data
- b. App\_Code
- c. App\_Themes
- d. App\_Browsers

**93. What is the difference between a page theme and a global theme?**

- A page theme is stored inside a subfolder of the App\_Themes folder of a project and applied to individual Web pages of that project. Global themes are stored inside the Themes folder on a Web server and apply to all the Web applications on the Web server.

**94. What is the difference between a default skin and a named skin?**

- The default skin is applied to all the Web server controls in a Web form, which are of similar type, and it does not provide a Skin ID attribute. The named skin provides a Skin ID attribute and users have to set the Skin ID property to apply it.

**95. Which of the following values does the <passwordFormat> attribute take?**

- a. Only Clear
- b. Only SHA1
- c. Only MD5
- d. Any one of these
- d. The <passwordFormat> attribute support all these values.

**96. What setting must be added in the configuration file to deny a particular user from accessing the secured resources?**

- To deny a particular user form accessing the secured resources, the web.config file must contain the following code:

```
<authorization>
  <deny users="username" />
</authorization>
```

**97. How can you enable impersonation in the web.config file?**

- To enable impersonation in the web.config file, you need to include the <identity> element in the web.config file and set the impersonate attribute to true as shown in the following code snippet:

```
<identity impersonate="true"/>
```

**98. Define a multilingual Web site.**

- A multilingual Web site serves content in a number of languages. It contains multiple copies for its content and other resources, such as date and time, in different languages.

**99. Differentiate globalization and localization.**

- The globalization is a technique to identify the specific part of a Web application that is different for different languages and make separate that portion from the core of the Web application. The localization is a procedure of configuring a Web application to be supported for a specific language or locale.

**100. What do you understand by the culture?**

- The culture denotes a combination of a language and optionally a region or a country. The contents of a Web page of a multilingual Web site are changed according to the culture defined in the operating system of the user accessing the Web page.

**101. What do you mean by a neutral culture?**

- When you specify a language but do not specify the associated country through a culture, the culture is called as a neutral culture.

**102. What are the Culture and UICulture values?**

- The Culture value determines the functions, such as Date and Currency, which are used to format data and numbers in a Web page. The UICulture value determines the resources, such as strings or images, which are loaded for a Web page in a Web application.

**103. What is the difference between page-level caching and fragment caching?**

- In the page-level caching, an entire Web page is cached; whereas, in the fragment caching, a part of the Web page, such as a user control added to the Web page, is cached.

**104. Suppose a Web page has been cached for 60 seconds and a user control added to this Web page has been cached for 30 seconds. After how much time, the cache for the user control expires.**

- a. 30 seconds
- b. 60 seconds
- c. 90 seconds
- d. 120 seconds
- b. The user control cache expires after 60 seconds.

**105. What is the difference between absolute expiration and sliding-time expiration?**

- The absolute expiration expires a cached item after the provided expiration time. The sliding time does not expire the cached items because it increments the specified time.

**106. If we have cached a dataset that has a sliding time expiration set as 20 seconds and we access this dataset in 18th second after being cached, the total time for which the dataset will present in the cache is:**

- a. 20 seconds
- b. 28 seconds
- c. 38 seconds
- d. 40 seconds
- c. The total time will be 38 seconds.

**107. Explain file-based dependency and key-based dependency.**

- In file-based dependency, you have to depend on a file that is saved in a disk. In key-based dependency, you have to depend on another cached item.
- Aggregate dependency allows multiple dependencies to be aggregated for content that depends on more than one resource. In

**108. What do you understand by aggregate dependency?**

such type of dependency, you need to depend on the sum of all the defined dependencies to remove a data item from the cache.

**109. What is the difference between adding items into cache through the Add() method and through the Insert() method?**

- Both methods work in a similar way except that the Cache.Add() function returns an object that represents the item you added in the cache. The Cache.Insert() function can replace an existing item in the cache, which is not possible using the Cache.Add() method.

**110. What is the difference between SQL notification and SQL invalidation?**

- The SQL cache notification generates notifications when the data of a database changes, on which your cache item depends. The SQL cache invalidation makes a cached item invalid when the data stored in a SQL server database changes.

**111. What is the difference between ASP session and ASP.NET session?**

- ASP does not support cookie-less sessions; whereas, ASP.NET does. In addition, the ASP.NET session can span across multiple servers.

**112. Explain the cookie less session and its working.**

- ASP.NET manages the session state in the same process that processes the request and does not create a cookie. It is known as a cookie less session. If cookies are not available, a session is tracked by adding a session identifier to the URL. The cookie less session is enabled using the following code snippet:

```
<%@SESSIONSTATE</%>
```

**113. How can you implement the postback property of an ASP.NET control?**

- You need to set the AutoPostBack property to True to implement the PostBack property of controls.

**114. What is AutoPostBack?**

- If you want a control to postback automatically when an event is raised, you need to set the AutoPostBack property of the control to True.

**115. How can you ensure that no one has tampered with ViewState in a Web page?**

- To ensure that no one has tampered with ViewState in a Web page, set the EnableViewStateMac property to True.

**116. How can you register a custom server control to a Web page?**

- You can register a custom server control to a Web page using the @Register directive.
- The @Page directive is responsible for this.

**117. How can you assign page specific attributes in an ASP.NET application?**

- The @Page directive is used to post a Web page to another Web page?

- The Response.Redirect method is used to post a page to another page, as shown in the following code snippet:

```
<%@PAGE LANGUAGE="C#" CODEFILE="Default.aspx.cs" %>
```

**118. What is cross-page posting in ASP.NET?**

- The server.Transfer() method is used to post data from one page to another. In this case, the URL remains the same. However, in cross page posting, data is collected from different Web pages and is displayed on a single page. To do so, you need to set the PostBackUrl property of the control, which specifies the target page. In the target page, you can access the PreviousPage property. For this, you need to use the @PreviousPageType directive. You can access the controls of previous page by using the FindControl() method.

**119. Why do you use the App\_Code folder in ASP.NET?**

- The App\_Code folder is automatically present in the project. It stores the files, such as classes, typed data set, text files, and reports. If this folder is not available in the application, you can add this folder. One of the important features of the App\_Code folder is that only one dll is created for the complete folder, irrespective of how many files it contains.

**120. What do you mean by the Web Part controls in ASP.NET?**

- The Web Part controls are the integrated controls, which are used to create a Web site. These controls allow the users to change the content, outlook, and state of Web pages in a Web browser.

**121. Write a code snippet to start a Notepad file in ASP.NET.**

- The following code snippet displays how to start a Notepad file:
- *Code for VB.NET:*

```
<%@PAGE LANGUAGE="VB" %>
```

- *Code for C#:*

```
<%@PAGE LANGUAGE="C#" %>
```

**123. What is ViewState?**

- The ViewState is a feature used by ASP.NET Web page to store the value of a page and its controls just before posting the page. Once the page is posted, the first task by the page processing is to restore the ViewState to get the values of the controls.

**124. Describe the complete lifecycle of a Web page.**

- When we execute a Web page, it passes from the following stages, which are collectively known as Web page lifecycle:

  - *Page request*—During this stage, ASP.NET makes sure the page either parsed or compiled and a cached version of the page can be sent in response
  - *Start*—During this stage sets the Request and Response page properties and the page check the page request is either a postback or a new request
  - *Page Initialization*—During this stage, the page initialize and the control's Unique Id property are set
  - *Load*—During this stage, if the request is postback, the control properties are loaded without loading the view state and control state otherwise loads the view state
  - *Validation*—During this stage, the controls are validated
  - *Postback event handling*—During this stage, if the request is a postback, handles the event
  - *Rendering*—During this stage, the page invokes the Render method to each control for return the output
  - *Unload*—During this stage, when the page is completely rendered and sent to the client, the page is unloaded

**125. What is the difference between the Response.Write() and Response.Output.Write() methods?**

- The Response.Write() method allows you to write the normal output, whereas, the Response.Output.Write() method allows you to write the formatted output.

**126. What is the lifespan for items stored in ViewState?**

- The items stored in ViewState live until the lifetime of the current page expires including the postbacks to the same page.

**127. Which of the following data bind controls does not support the sorting feature?**

- a. GridView control
- b. Repeater control

- |                               |                            |
|-------------------------------|----------------------------|
| <b>c. DetailsView control</b> | <b>d. DataList control</b> |
|-------------------------------|----------------------------|

**128. Make a list of all templates of the Repeater control.**

- ItemTemplate
- AlternatingItemTemplate
- SeparatorTemplate
- HeaderTemplate
- FooterTemplate

**129. Which of the following templates cannot be skipped in the Repeater control?**

- |                             |                                   |
|-----------------------------|-----------------------------------|
| <b>a. ItemTemplate</b>      | <b>b. AlternatingItemTemplate</b> |
| <b>c. SeparatorTemplate</b> | <b>d. FooterTemplate</b>          |

- a. The ItemTemplate cannot be skipped in the Repeater control.

**130. Write three common properties of all validation controls.**

- Three common properties of validation controls are as follows:

  - *ControlToValidate*—Provides a control to validate
  - *ErrorMessage*—Displays an error message
  - *IsValid*—Specifies if the control's validation has succeeded or not
  - *Text*—Displays a text for validation control before validation

**131. Can we validate a DropDownList by RequiredFieldValidator?**

- Yes, we can validate a DropDownList by RequiredFieldValidator. To perform this validation, we have to set the InitialValue property of RequiredFieldValidator control.

**132. Explain login controls.**

- Login controls are built-in controls in ASP.NET for providing a login solution to ASP.NET application. The login controls use the membership system to authenticate a user credentials for a Web site. There are many controls in login controls.

  - *ChangePassword control*—Allows users to change their password.
  - *CreateUserWizard control*—Provides an interface to the user to register for that Web site.

- Login control*—Provides an interface for user authentication. It consists of a set of controls, such as TextBox, Label, Button, Checkbox, Hyperlink.
- LoginView control*—Displays appropriate information to different users according to the user's status.
- LoginStatus control*—Shows a login link to users, who are not authenticated and logout link, who are authenticated
- LoginName control*—Displays a user name, if the user logs in.
- PasswordRecovery control*—Allows users to get back the password through an e-mail, if they forget.

**133. What are navigation controls? How many navigation controls are there in ASP.NET 4.0?**

- Navigation controls help you to navigate in a Web application easily. These controls store all the links in a hierarchical or drop-down structure; thereby facilitating easy navigation in a Web application. There are three navigation controls in ASP.NET 4.0.
  - SiteMapPath
  - Menu
  - TreeView

**134. Can you set which type of comparison you want to perform by the CompareValidator control?**

- Yes, by setting the Operator property of the CompareValidator control.

**135. How can we provide the WebParts control functionality to a server control?**

- We can provide the WebParts controls functionality to a server control by setting the CreateWebPart property of WebPartManager.

**136. Which control is responsible to create a space on Web pages?**

- a. WebPartManager control
  - b. WebPartZone control
  - c. EditorZone control
  - d. PageCatalogPart control
- b. The WebPartZone control is responsible to create a space on the Web page.

**137. Which is the parent class of the Web server control?**

- The System.Web.UI.Control class is the parent class for all Web server controls.

**138. What is difference between a Label control and a Literal control?**

- The Label control's final HTML code has an HTML tag; whereas, the Literal control's final HTML code contains only text, which is not surrounded by any HTML tag.

**139. What is the use of Placeholder control? Can we see it at runtime?**

- The Placeholder control acts as a container for those controls that are dynamically generated at runtime. We cannot see it at runtime because it does not produce any visible output. It used only as a container.

**140. Explain the AdRotator Control.**

- The AdRotator is an ASP.NET control that is used to provide advertisements to Web pages. The AdRotator control associates with one or many advertisements, which randomly displays one by one at a time when the Web page is refreshed. The AdRotator control advertisements are associated with links; therefore, when you click on an advertisement, it redirects you to other pages. The AdRotator control is associated with a data source, which is normally an XML file or a database table. A data source contains all the information, such as advertisement graphics reference, link, and alternate text. Therefore, when you use the AdRotator control, you should first create a data source and then associate it with the AdRotator control.

**141. Which method has been introduced in ASP.NET 4.0 to redirect a page permanently?**

- The RedirectPermanent() method added in ASP.NET 4.0 to redirect a page permanently. The following code snippet is an example of the RedirectPermanent() method:

```
Response.RedirectPermanent("http://www.google.com")
```

**142. Which two new properties are added in ASP.NET 4.0 Page class?**

- The two new properties added in the Page class are MetaKeyword and MetaDescription.

**143. What is the function of the ViewState property?**

- The ASP.NET 4.0 introduced a new property called ViewStateMode for the Control class. Now you can enable the view state to an individual control even if the view state for an ASP.NET page is disabled.

**14.4. Which are two new controls introduced in ASP.NET 4.0. Explain them.**

- The Chart control and the QueryExtender control are two new controls in ASP.NET. The Chart control includes simple charts for statistical and financial analysis. It helps the developers to create rich and professional looking data visualization solutions. It includes a wide range of chart options, such as pie, area, range, point, circular, accumulation, data, distribution, and doughnut.

The QueryExtender is used to filter and sort data that is retrieved from data source. Now, you do not need to use a Where clause explicitly while retrieving the data from data source. It works only with LinqDataSource and EntityDataSource controls, so if you use these controls with other data source controls, it will not respond anything. QueryExtender supports various type of filtering expression, such as SearchExpression, RangeExpression, PropertyExpression, OrderByExpression, and CustomExpression.

**14.5. List the features of the Chart control.**

- The following are the features of the Chart control:

- Bounds a chart with any data source
- Simple manipulation of chart data, such as copying, merging, grouping, sorting, searching, and filtering
- Support many statistical and financial formulas for data analysis
- Provide advanced chart outlook, such as 2-D, 3-D, lighting, and perspective
- Support events and customizations
- Includes interactivity with Microsoft AJAX
- Supports AJAX Content Delivery Network (CDN)

# 10

## Windows and Web Services

**1. What are Windows services?**

- Windows services, previously known as NT services, are applications that are installed on the system as system services. In other words, Windows services are applications that run in the background with the Windows operating system. The primary use of Windows services is to reduce the consumption of memory required for performing backend operations. Let's take an example to understand this easily.

Suppose you want to perform a variety of functions, such as monitor the performance of your computer or application, check the status of an application, and manage various devices, such as printers. In such a case, you can use Windows services to reduce memory consumption. In addition, Windows services can run on your system even if you have not logged on to your computer. In addition, these services do not have any user interface.

**2. Can you share a process between Windows services?**

- Yes, you can share a process between Windows services.

**3. In .NET, which is the parent class to create all Windows services?**

- The ServiceBase class is the parent class to create all Windows services.

**4. Which class in .NET is used to install a Windows service?**

- The ServiceInstaller class, also known as the project installer class, is used to install a Windows service.

**5. While installing a Windows service, an EventLogInstaller class is automatically created to install the event log related to the particular service. Is it true?**

- Yes, it is true.

**6. Which tool installs the DLL on your local computer and installs the Windows service in a transactional manner?**

- The Installutil.exe tool.

**7. Which property of the ServiceBase class can be used to specify whether a service can be paused and resumed?**

- The CanPauseAndContinue property provides such type of service.

**8. Which method is used to uninstall the Windows services?**

- The Uninstall() method is used to uninstall the Windows services.

**9. Describe the EventLog class.**

- The EventLog class is used to access the Windows event logs from Windows services. Using EventLog, you can also customize Windows event logs that record information about important software and hardware events, such as the events of the .NET controls, keyboard, or other hardware devices. The EventLog class allows you to read or write to event logs, delete logs, and create as well as delete event sources. You can use the EventLog class to create event logs while creating an event source. An event source can be used to write to only one event log at a particular time. However, it is possible to associate one event log to multiple sources.

**10. Explain the concept of Web services in brief.**

- A Web service may be defined as an independent and self-sustained unit of a software application that is hosted on the Web and implements specific functionalities to execute the business logic. A Web service provides so many functionalities, such as generating pay slips for employees, computing tax, broadcasting weather report, and providing updated news. The Web service allows application to share information or exchange data with other applications across different operating systems and hardware. Therefore, the work of a Web service is to unite software by exchanging data irrespective of their operating systems, supported hardware, and programming language used in their development. The Web services transfer data in the XML format and use Simple Object Access Protocol (SOAP) to communicate. It is an XML based protocol. The Web services use Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI) to describe itself.

**11. What advantages have Web services over Component Object Model (COM) and Distributed Component Object Model (DCOM)?**

- The advantages of Web services over COM and DCOM are as follows:
  - Web services are simple to use and can be implemented on varied platforms
  - Web services are loosely coupled; as a result, their interfaces and methods can be extended
  - Web services do not carry any state information with them so that multiple requests can be processed simultaneously

**12. Write the file extension for a Web service.**

- A Web service file extension is .asmx file. For example, Service1.asmx is a Web service file.

**13. Which two methods are used to discover the URL of Web services?**

- The two methods to discover the URL of Web services are Web service discovery tool (Disco.exe) and UDDI.

**14. How do Web services communicate with applications across diverse platforms and hardware?**

- Web services use Simple Object Access Protocol (SOAP) in order to expose the business functionalities to the applications. The SOAP uses XML to create the standardized structure of data, which can be exchanged between two entities by the standard protocols, such as Hypertext Transfer Protocol (HTTP). As XML is a platform and hardware neutral format; the SOAP protocol enables the Web to exchange information across platforms and devices. Therefore, the customer application of a Web service is shielded completely. It means that you cannot implement any details about the XML over HTTP. Other platforms, such as Linux and UNIX can use Web services that are hosted on Windows.

**15. What is the structure of a SOAP message that is used by Web services for communication?**

- The SOAP protocol uses two distinct structures to organize a message for exchanging information through Web services. These structures are SOAP header and the SOAP body. A SOAP header is appended with a SOAP body to form a complete SOAP message.

The SOAP header contains the control information, which is used by the consumer application of a Web service. Control information is

additional to the main message and is generated by the SOAP protocol itself. The SOAP headers contain the following information:

- Information about transaction
- Information about various types of authentication services
- Information about culture such as language, country, currency
- Information about the next recipient of the message
- State information to be carried between requests

The SOAP body contains the actual message that is intended for the recipient application.

#### 16. What do you understand by SOAP encoding?

The serialization of the types, such as integers and strings, inside a SOAP message is called encoding. The SOAP objects use XML elements and attributes to serialized data. For example, encodingStyle is an attribute of the Envelop element, which is used to specify the encoding rules for a SOAP object.

#### 17. What is the use of a .disco file?

A client application uses a .disco file to locate or discover the documents that contain the description of a Web service. The .disco file contains links to other resources, which describe essential features, such as capabilities of a Web service. The links contained in a .disco file can refer to other discovery documents or XSD schemas. The description about the services and capabilities of a Web service is written in Web services Description Language (WSDL). A .disco file can also contain the information about other XML Web services that reside on the same or a different Web server.

#### 18. Which namespace must be included in a code that enables a XML Web service to write events in an event log file?

The System.Diagnostics is the namespace, which must be included in a code to enable a Web service for writing events in an event log file.

#### 19. What is Universal Description, Discovery, and Integration (UDDI)?

UDDI is a centralized directory where Web services offered by different organizations are published. It is a platform-independent method that is built into the .NET platform. A UDDI directory has an advantage over a DISCO file in that the directory provides a single location where clients can search the Web services offered by different organizations.

#### 21. Explain the WSDL.

WSDL is a short form for Web Services Description Language, which is used to describe a Web service in terms of the messages that it creates and accepts. The WSDL document is an XML file that contains the interface schema for the Web service. It identifies the methods that are used during the exchange between a Web service consumer and a Web service provider. The following are the elements contained in the WSDL document:

- Types* — Describes the variations of data types that are used to exchange messages between the user and the provider
- Message* — Describes the actual message or method call
- portType* — Describes the set of operations and each related message

The following are some of the important features of UDDI:

- Acts as a directory and stores information about Web services
- Uses the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) Internet standards, such as Extensible Markup Language (XML), Hypertext Transfer Protocol (HTTP), and Domain Name Server (DNS) protocols
- Uses WSDL to describe interfaces to Web services
- Communicates through SOAP

Some benefits of using UDDI are:

- Helps to search the right business organization from millions of business organizations that are currently online
- Enables commerce, once the preferred business is discovered
- Helps to reach new customers and increase access to current customers
- Helps to expand offerings and market reach
- Allows rapid participation in the global Internet economy
- Describes Web services and business processes programmatically in a single, open, and secure environment

- binding**—Describes the protocol details
- service**—Used to make groups a set of related ports together
- 22. Which step is necessary to perform before a Web service can be consumed?**
- It is necessary to build a proxy class by using the wsdl.exe utility before a Web service can be consumed.
- 23. Write the names of public properties defined in the WebService class.**
- There are many properties defined in the WebServices class:
    - Application**—Obtains the application object for the current HTTP request
    - Context**—Obtains the HttpContext object for the current request, which encapsulates all HTTP-specific context used by the HTTP server to process Web requests
    - Server**—Obtains the HttpServerUtility object for the current request
    - Session**—Obtains the HttpSessionState object for the current request
    - SoapVersion**—Obtains the version of the SOAP protocol used to make the SOAP request to a Web service
    - User**—Obtains the Server User Object. This property can be used to authenticate whether a user is authorized to execute the request.
- 24. Mention the name of the directory where it is necessary to locate the proxy file to use a Web service.**
- The proxy file must be stored in the /bin directory. This directory is situated under the root directory of the application.
- 25. Mention the namespace that you must import in code to build a Web service.**
- System.Web.Services is the elementary namespace, which must be imported to develop code of a Web service.
- 26. Give a description of the infrastructure and the Web resources that a client application uses to consume the services provided by a Web service.**
- The infrastructure and the Web resources that are used by client application to consume the services provided by a Web service can be explained with the help of a schematic diagram. Figure 10.1

shows how various components of the XML Web services infrastructure enable clients to locate and call methods on XML Web services:

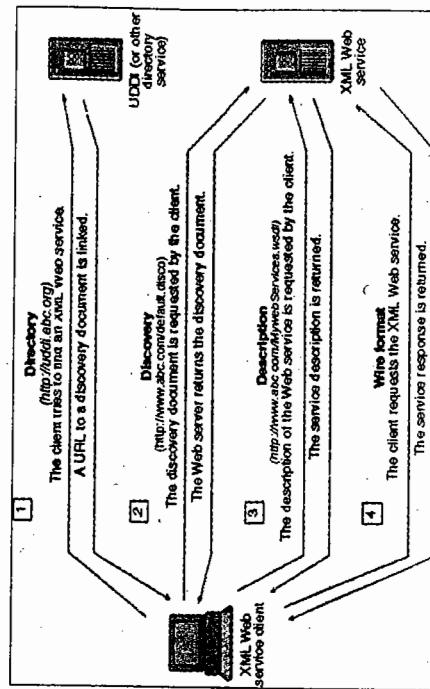


Figure 10.1: Components of XML Web Service Infrastructure

When a client accesses a UDDI service to locate an XML Web service, the UDDI service returns a URL to the discovery document of the XML Web service. A discovery document is a .disco file, which contains the link to the resources that describe an XML Web service. A discovery file is an XML document that enables programmatic discovery of an XML Web service. After the client receives the URL to the discovery document, the client requests a server, which returns the discovery document to the client. The content of a sample discovery document is shown in the following code snippet:

```

<?xml version="1.0"?>
<discovery xmlns="http://schemas.xmlsoap.org/disco/>
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:contractRef ref="urn:abc:wsdl:abc:wsdl:disco:discovery">
  
```

The client uses the information in the discovery document and requests a server to return the service description of an XML Web service. The service description is a .wsdl file and enables a client to interact with an XML Web service.

- 27. How does a client application communicate with an XML Web service?**

- A client application uses proxy objects to communicate with an XML Web service.
- Use of proxy object for communication with Web service

is similar to the use of a Remote Procedure Call (RPC). The proxy object of XML Web service is located on the local computer, that is, clients machine.

Figure 10.2 shows various stages in a communication process that occurs between a client and an XML Web service:

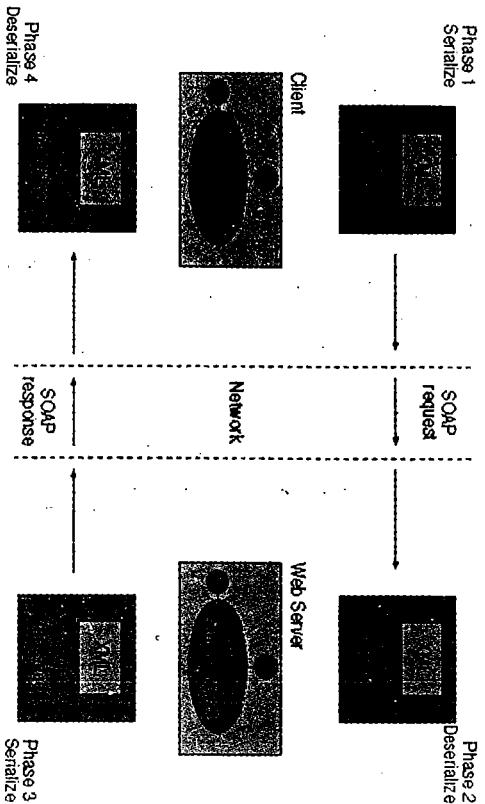


Figure 10.2: Web Services Communication Architecture

As shown in Figure 10.2, the interaction between a client and an XML Web service consists of several phases. The following tasks are performed during these phases:

- The client makes an instance of the XML Web service proxy class on the client computer.
- The client application invokes a function on the proxy object.
- The XML Web services infrastructure encodes the functions call and arguments into a SOAP message on the client system and post it to the XML Web service over the network.
- The structure on the server where XML Web services store decodes the SOAP message and creates an instance of the XML Web service. The structure then invokes the function with the arguments on the XML Web service.
- The XML Web service executes the function and returns the value by out parameters to the structure.
- The structure encodes the return value and out parameters into a SOAP message and posts them to the client over the network.

- The structure on the client system de-codes the SOAP message that consists of the return value and out parameters and sends them to the proxy object.
- The proxy object can send the return value to the client system.
- To create an XML Web services that the clients can use, you should use the ASP.NET infrastructure. It is a part of .NET Framework. The Visual Studio .NET provides the type of environment to develop, publishes, and deploys XML Web services using ASP.NET.

#### 28. Which attribute is used to create the Web method?

- The Web service consists of one or more public methods, which are preceded by the WebMethod attribute. This attribute explains the functionality of a Web service to client applications.
- You can declare a Web method in .NET by using the following code snippets:

In C#:

```
[%WebMethod%]
```

In VB:

```
<%@WebService Language="VB"
CodeBehind="AppCode\Service.vb"
Class="Service"%>
```

#### 29. What are the Page directives of a Web service's code?

- The @WebService directive is a Page directive to the Web service's code. It is contained in the Service.asmx file of Web service application. The following code snippets show the @WebService directive in Web service projects:

In VB:

```
[%@WebService Language="VB"
CodeBehind="AppCode\Service.vb
Class="Service"%>
```

In C#:

```
[%@WebService Language="C#
CodeBehind="AppCode\Service.cs
Class="Service"%>
```

The simple Web service directive has three attributes. The following list explains these attributes:

- **Class** —Specifies the class name that consists of the Web methods of the Web service.
- **CodeBehind** —Specifies code-behind file name with .vb or .cs extension. It consists of the business logic that is required for the Web methods.

- Language**—Specifies the language, which you are using for that Web service

**30. Which property of the WebMethod attribute allows you to maintain the state of objects across sessions in a Web method?**

- The WebMethod attribute's EnableSession property enables you to enable session state for a Web method.

**31. What does a Session property do?**

- The Session property of the WebService class is used to obtain the HttpSessionState instance for the current request. The syntax of this property is shown in the following code snippets:

In VB:

```
VB.NET:    Session("CustomerID") = Session("CustomerID")
          Session("CustomerID") = Session("CustomerID")
          Session("CustomerID") = Session("CustomerID")
```

In C#:

```
C#:    Session["CustomerID"] = Session["CustomerID"];
      Session["CustomerID"] = Session["CustomerID"];
      Session["CustomerID"] = Session["CustomerID"]
```

**32. What steps do you need to perform to enable transactions in a Web method?**

- Perform the following steps to allow transactions in a Web method:
  - Add the System.EnterpriseServices.dll file reference using solution explorer
  - Import the System.EnterpriseServices namespace into the XML Web service application
  - Set the TransactionOption property to an appropriate value

**33. How can you ensure that only authorized users access your Web service?**

- You should use the <authorization> element to ensure that only authorized users access your Web service. This element allows or denies access to your Web service according to their role.

**34. Does a Web service have state?**

- The Web services do not have any technique to maintain state. However, it can access ASP.NET objects, such as application and session if they extend from the WebService base class.

**35. What is the use of the mustUnderstand attribute in the Header element of a SOAP message?**

- The mustUnderstand attribute indicates that a header entry is either required or optional for the recipient to process further.

**36. What does the portType element of a WSDL document contain?**

- The portType element contains the operations exposed by the Web service, and the messages involved in the communication between the Web service and its consumers.

**37. What is DISCO?**

- DISCO is a technology developed by Microsoft to publish and discover Web services. It discovers URLs of all XML Web services located on a Web server and creates a list of these Web services in a file called as a DISCO file.

**38. What advantage UDDI has over DISCO?**

- The UDDI directory has an advantage over a DISCO file, as it provides a single location where a client can find the Web services offered by different organizations.

**39. Explain the HTTP-GET and HTTP-POST protocol.**

- The HTTP-GET protocol consists of a query string to send the request parameters to the server. The GET request uses the ScriptMethodAttribute attribute to call a configured method of a Web service. The size of the data to be transmitted is restricted to the length of the URL allowed by the client's browser.

The HTTP-POST protocol consists of a body containing the data that the client's browser sends to the server. There is a restriction on the amount of data to be transmitted. Therefore, whenever the transmitted data exceeds the size limit of the GET request, you can use the POST request. The transmission of data starts from the client's browser, when the client serializes the request into the JSON format and sends it as the POST data to the server. When the server gets the request, it first de-serializes the data in the JSON format into

the .NET Framework compatible type and then makes the actual call to the Web service. In response, the server serializes the return values and passes them back to the client, and the client de-serializes these values into JavaScript objects for processing.

#### 40. How can you prevent your Web services from unauthorized access?

The following are the ways to prevent your Web service from unauthorized access:

- Using encryption and message-based security
- Using authentication and access controls for the Web service

# 11

## Windows Workflow Foundation

### 1. What is Windows Workflow Foundation (WF)?

Windows Workflow Foundation (WF) is a technology that was first introduced in .NET Framework 3.0. WF consists of a programming model, a workflow runtime engine, workflow designer, a rules engine, and tools to quickly build workflow-based applications on Windows. WF facilitates the separation between the business process code and the actual implementation code.

### 2. What are the components of WF 4.0?

WF consists of several components that work together to create a desired workflow. The components of WF are given as follows:

- Workflows and activities
- Base activity library
- Custom activities
- Host process
- Activity data model
- Runtime engine
- Runtime services

### 3. What is a workflow?

A workflow is a collection of actions (called activities) that presents the model of a process. A workflow provides a way to describe the order of the execution of a long running process and relationships between different activities. Multiple instances of a workflow may be active at any given moment in an application.

### 4. What are the different types of workflow in WF?

- In WF 4.0, the following two types of workflows are used:
  - Flowchart workflows* —Helps you to create workflows using the common flowchart elements. In WF, the Flowchart activity is generally used to implement a non-sequential workflow, and occasionally it implements sequential workflows in case the **FlowDecision** nodes are not used. The Flowchart activity

- contains a collection of flow nodes, which inherit from the `FlowNode` class. The following types of nodes or elements can be a part of a flowchart:
- ***FlowStep***—Executes activities of a flowchart in a sequence.
  - ***FlowDecision***—Shows the execution on the basis of a Boolean condition. It is similar to the If construct.
  - ***FlowSwitch***—Shows the execution on the basis of an exclusive switch. It is similar to the Switch construct.
  - ***Procedural workflows***—Helps you to create workflows using basic and sequential execution standards. In WF, procedural workflows use flow control constructs, such as While, Switch, ForEach, and If, to execute activities. These flow control constructs are similar to those found in procedural languages. Procedural workflows can also contain other flow control activities, such as Flowchart and Sequence.

#### 5. What are the four workflow principles?

- According to Microsoft, there are four major principles that explain the behavior and working of workflows. Developers can use these principles while developing workflow-based applications. The four principles are as follows:
  - Workflows help in coordinating the work performed by people and software
  - Workflows are long-running and stateful
  - Workflows are based on extensible models
  - Workflows remain transparent and dynamic throughout their lifecycle

#### 6. What is an activity?

- In Windows Workflow Foundation 4.0, an activity is the basic unit of composition and execution of a workflow. Each activity in a workflow consists of its own variables and arguments and is a subclass of the `Activity` class. These activities provide facilities for flow control, exception handling, data consistency, loading or unloading workflows, tracking, and transaction flow.

#### 7. What is the difference between a system workflow and a human workflow?

- A system workflow is a workflow that is developed to automate interactions among applications. Such workflow is usually static and predictable. On the other hand, a human workflow is a workflow that coordinates interactions of applications with people. As human workflows involve both software and people, they need to be more flexible than system workflows.

#### 8. What is a host process?

- A host process is an executable program that hosts a workflow. It may be a Windows Forms application, a Web application, or a Web service application. You can use Web services in the host process or remoting to enable other applications to communicate with the workflow.

#### 9. What is a runtime engine?

- A runtime engine of WF provides the basic functionality to execute and manage the workflow lifetime. It runs within the host process and is responsible for executing each workflow instance. A host process can interact with multiple runtime engines at a time, where each engine executes multiple workflow instances. The host process interacts with runtime engine by using any of the following classes:
  - ***WorkflowInvoker***—Invokes a workflow as its method
  - ***WorkflowApplication***—Controls the execution of a single workflow instance explicitly
  - ***WorkflowServiceHost***—Hosts the workflows and allows sending and receiving messages among various instances of workflows

#### 10. What are runtime services?

- Runtime services consist of predefined and user-defined classes that are available to the workflow runtime engine during execution to customize the behavior of workflow runtime. Some of the runtime services available in WF 4.0 are as follows:
  - ***Scheduling services***—Enable creating and scheduling new workflow instances for execution.
  - ***Work batch services***—Enable behavior to maintain a stable and consistent execution environment.
  - ***Persistence services***—Enable you to save or restore the state of a running workflow for later use. You can restart the saved workflow anytime in future, even after weeks of inactivity.
  - ***Tracking services***—Enable you to monitor the state of the workflows. This is particularly useful when you have multiple workflows active at the same time (for example, in a shopping cart application).
  - ***Timer service***—Manages the timing required by the `DelayActivity` activity.
  - ***Transactions services***—Provide the transaction support needed for data integrity.
  - ***Data exchange services***—Manage custom communication services.
  - ***Threading services***—Administer physical threads used to execute workflow instances.

### 11. Explain the mygmailpasswordactivity data model of WF 4.0.

- In WF 4, activities store and maintain the data by using the following elements:
  - *Variable*—Stores the data in an activity
  - *Argument*—Moves the data in and out from an activity and a workflow
  - *Expression*—Represents an activity that returns the values used in arguments

### 12. What are XOML files?

- WF provides developers a declarative way to create workflows by using extensible Application Markup Language (XAML). The files used to store such workflow markups are known as extensible Object Markup Language (XOML) files.

### 13. What is a dynamic update?

- Dynamic update is a powerful feature of WF that describes the ability of WF to modify the execution path of a running workflow. This feature is used in circumstances that call for extraneous behavior that was not modeled by the original workflow developer.

### 14. What is a base activity library?

- The base activity library is a collection of activities used to create workflows.

### 15. Explain Custom Activities.

- In addition to the standard activities available within the base activity library, you can create new activities to meet specific business needs. Creating custom activities may be required to support a particular application that you want to integrate with WF. Custom activities are generally created through attributes and inheritance. You can create two types of custom activities, base and composite. You can create basic custom activity by inheriting the `Activity` class and custom composite activity by inheriting the `CompositeActivity` class or a derived type.

### 16. Explain why workflows are based on Extensible Models.

- Workflows serve the purpose of automating business processes. Now, since each type of business has a wide range of problems; therefore, a workflow platform needs to be extensible. WF provides you with a set of base activities, such as IfElse, Code, and Delay, to build a workflow. You can extend these activities or build new activities to meet your requirements. Besides activities, you can also extend services, such as tracking, management, and persistence, provided by the runtime engine.

### 17. What are the new features in WF 4.0?

- The following are the new features of WF 4.0:
  - *Improved workflow designer*—Supports complex workflows and provides easier mechanisms to create custom activities. You can use XAML to define the appearance of your activities and the way the activities interact with users in a visual designed environment. In addition, you can re-host the workflow designer in your applications to enable other users to easily view and interact with your workflows. Re-hosting a workflow designer also provides a customized user experience for workflow design.
  - *Improved data flow model*—Provides two options, variables and arguments, for data storage and data flow. You can use variables to store the data in an activity. You can also use some specialized arguments, such as **InArgument**, **InOutArgument**, and **OutArgument**, to determine the direction of data while moving the data in and out of the activity.
  - *The Flowchart control/flow activity*—Enables you to implement loops and conditional branching in workflows. The Flowchart control activity allows you to create flexible workflows in which you can navigate back to previous steps as well as skip steps based on conditions within the workflow. You can add the Flowchart activity into your workflow either by the code or by using the Workflow designer. You can create multiple Flowchart activities in a workflow, but execute only one activity at a time.
  - *Workflow activity model*—Enables you to break complex business processes of a workflow into smaller processes. This ensures reliable flow control with automatic exception handling.
  - *Expanded built-in activity library*—Introduces the following new activities:
    - New flow control activities, such as DoWhile, Pick, TryCatch, ForEach, Switch, and ParallelForEach
    - Activities for manipulating data, such as Assign
    - Collection activities, such as AddToCollection
    - Activities for controlling transactions, such as TransactionScope and Compensate
    - Messaging activities, such as SendContent and ReceiveReply
  - *Bookmark object*—Enables you to resume a pending workflow.
- In WF 4.0, a bookmark is a mechanism that enables an activity to wait for an input without interrupting a workflow thread. When an activity signals that it is waiting for the input from a user, it can create a bookmark. A bookmark is created by using the `BookmarkOptions` class. This class provides the following bookmark types:

- None* —Represents a bookmark that can be resumed exactly once. This is the default bookmark type.
- MultipleResume* —Refers to a bookmark that you can resume multiple times.
- NonBlocking* —Refers to a bookmark that does not block the functioning of the workflow.

19. Which component of WF architecture is responsible to execute each workflow instance?
- WF runtime engine is responsible to execute each workflow instance.
20. Write the steps that are involved in the sequential workflow, by default.
- By default, a sequential workflow has only two steps:
    - Start
    - Finish

21. How can you implement a condition in a workflow?

- You can implement a condition by using either of the following ways:
  - By creating a rule condition* —Specifies that you can implement conditions either directly in code or by using a tool, called the **Rule Condition Editor**. Rule conditions are stored in a separate Extensible Markup Language (XML) file. When a rule condition occurs in a workflow, the expression in a condition is evaluated and a **Boolean** value is returned.

- By creating a code condition* —Refers to defining a condition directly in code. A code condition can be created by writing a method in the code. The method contains code for the condition and returns a **Boolean** value.

22. What is the function of the Rule Condition Editor dialog box in WF?
- You can create and modify declarative rule conditions by using the **Rule Condition Editor** dialog box.

23. Which option do you need to select for the Condition property, if you want to create a code condition?
- You can select the **Code Condition** option to create a code condition.

24. Can you integrate workflow applications with some other application, such as Windows Forms applications and Web applications?
- Yes.

# 12

## Windows Presentation Foundation and Silverlight

1. What is the main advantage of using Windows Presentation Foundation (WPF) over Windows Forms?

- In .NET Framework 4.0, WPF enables a developer to develop graphic-rich desktop and browser-hosted applications. The main advantage of WPF, previously known as Avalon, is that it provides built-in support for different media, such as images, text, audio, video, 2-D and 3-D graphics, and different documents. WPF does not require any additional Application Programming Interface (API) while Windows Forms applications require various APIs, such as GDI+, Windows Media Player, and DirectX, to work with graphics, multimedia, and documents. With the introduction of WPF, it has become easy and simple to develop graphic-rich desktop applications.

2. When was WPF introduced and what is its latest version?

- WPF was introduced with .NET Framework 3.0 in 2006, previously called WinFX. The latest version of WPF is WPF 4.0, which is included in .NET Framework 4.0.

3. Explain the architecture of WPF 4.0.

- WPF 4.0 works on Extensible Application Markup Language (XAML).
- The designers use this language to create a User Interface (UI) in WPF. The WPF architecture consists of several components, such as PresentationFramework, PresentationCore, CLR, mscore, User32, DirectX, and Kernel. Figure 12.1 shows the architecture of WPF.

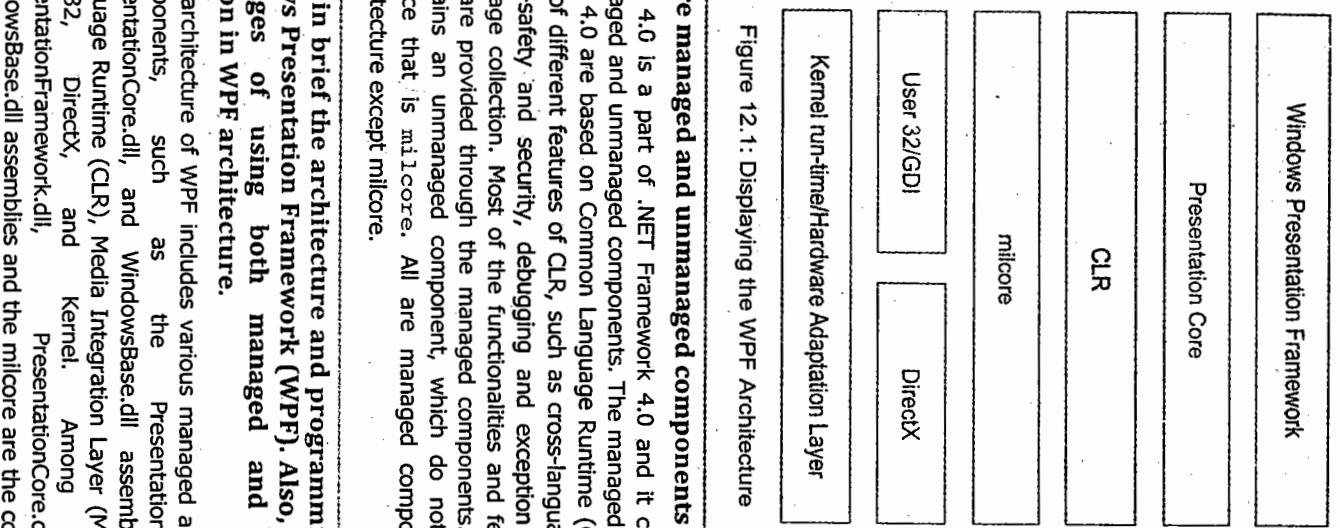


Figure 12.1: Displaying the WPF Architecture

#### 4. What are managed and unmanaged components in WPF 4.0?

- WPF 4.0 is a part of .NET Framework 4.0 and it consists of both managed and unmanaged components. The managed components of WPF 4.0 are based on Common Language Runtime (CLR) and make use of different features of CLR, such as cross-language integration, type-safety and security, debugging and exception handling, and garbage collection. Most of the functionalities and features of WPF 4.0 are provided through the managed components. WPF 4.0 also contains an unmanaged component, which do not use the CLR service that is milcore. All are managed components in WPF architecture except milcore.

#### 5. Discuss in brief the architecture and programming model of Windows Presentation Framework (WPF). Also, mention the advantages of using both managed and unmanaged execution in WPF architecture.

- The architecture of WPF includes various managed and unmanaged components, such as the PresentationFramework.dll, PresentationCore.dll, and WindowsBase.dll assemblies, Common Language Runtime (CLR), Media Integration Layer (ML or milcore), User32, DirectX, and Kernel. Among these, the PresentationFramework.dll, PresentationCore.dll, and WindowsBase.dll assemblies and the milcore are the components that

#### 6. What is the milcore?

- The milcore is an unmanaged component in WPF. It is the foundation of Media Integration Layer (ML); that is why, it is called milcore. This layer provides a complete replacement of Win32 rendering and performs the task of drawing pixels. The milcore layer helps to keep Central Processing Unit (CPU) free for other processing tasks by using computer's graphic card as much as possible.

#### 7. Explain the WPF features.

- The goal of introducing WPF is to provide a UI that integrates capabilities of previous frameworks including Graphical Device Interface (GDI), GDI+, and HyperText Markup Language (HTML). WPF uses the principle of managed programming model as the programmable platform. It supports XAML, which is a rich, consistent, and complete markup-based language. WPF provides the following advanced features:
  - **Documents**—Provides support for XML Paper Specification (XPS) and flow documents. The XAML FixedDocument tag supports XPS documents.
  - **Graphics**—Provides support for 2-D and 3-D graphics.
  - **Images**—Uses the image tag of XAML to support different type of image formats, such as jpeg, gif, and png.
  - **Media**—Uses the MediaElement tag of XAML for displaying video and audio formats, such as WMV, AVI, and MPEG.
  - **Animation**—Provides support for animation.
  - **Data Binding**—Provides support for data binding, which is automatically performed in any WPF application.

play a major role in WPF applications. PresentationFramework.dll, PresentationCore.dll, and WindowsBase.dll are the managed components and contribute in the managed programming model of WPF, while the milcore is an unmanaged component and contributes in the unmanaged programming model of WPF. A major portion of the programming model of WPF is managed as the execution of code is controlled by the CLR. This implies that these three managed components can leverage different CLR features, such as type safety, cross-language integration, and garbage collection. However, milcore is not managed by the CLR, rather it acts as an interface between the CLR and the DirectX components. Milcore is integrated firmly with the DirectX component to work with graphics, animations, and multimedia in WPF applications. The composition engine of the milcore uses the hardware and software rendering offered by DirectX and improves the performance of WPF applications.

### 8. What are the new improvements introduced in WPF 4.0?

- The following are the list of improvements that are introduced in WPF 4.0:
  - Visual State Manager*—Refers to the class introduced with WPF 4.0 that enables the .NET developers to define the appearance of a control according to the state of control.
  - Addition of new controls*—Refers to the introduction of new controls that include DataGrid, Calendar, and DatePicker.
  - Touch input and their manipulation*—Refers to the support for touch input that WPF 4.0 elements provide. UIElement, UIElement 3D, and ContentElement are classes, which consist of events that are raised when you touch an element on a touch screen.
  - New text rendering stack*—Refers to the improvements made in rendering text, which include text clarity, configurability, and support for international languages.

### 9. Name some of the important classes of WPF 4.0.

- The following are some of the important classes in WPF 4.0:
  - System.Windows.UIElement*—Enables you to work with the WPF controls and UI elements. The services provided by this class are managing the layout and handling events raised on the UI elements and controls.
  - System.Windows.FrameworkElement*—Refers to the class that acts as the interface between the UIElement class and the classes that implement the UI elements and controls. In addition to this, it also offers data binding and styles features.
  - System.Windows.Controls.Control*—Refers to the parent class for all WPF controls. This provides the common functionality to all controls, such as providing background color, add borders, and applying styles.
  - System.Windows.VisualStateManager*—Refers to the class introduced with WPF 4.0 that enables the .NET developers to define the appearance of a control. In other words, the Visual State Manager class can be used to apply different visual states to a control. For instance, if there is a button in your application, then using the Visual State Manager class, you can define how the button would appear if a user hovers the mouse over the button or presses the button. The Visual State Manager class is fully supported in Microsoft Expression Blend.

### 10. What are the different types of applications that you can create in WPF using Visual Studio 2010?

- Visual Studio 2010 provides project templates to create WPF applications. There are primarily two types of WPF applications:
  - Standalone applications*—Refer to applications that work as a standalone desktop application. These applications are same as the Windows Forms applications that you have to install on every client. The center point of designing a standalone application is a window, which is an object of the Windows class.
  - XAML browser applications (XBAPs)*—Refer to applications that are hosted on server and are accessed through a browser, such as Internet Explorer and Netscape. You do not require to install these applications on the client system. The XBAPs are same as the ASP.NET Web application designing and working.

### 11. Explain the main elements of WPF Designer.

- WPF Designer is the designer interface of WPF applications in Visual Studio 2010. WPF Designer has several elements that offer an easy-and-quick-to-design UI for WPF applications. The main elements of the WPF Designer in Visual Studio 2010 are as follows:
  - Design view*—Provides the core designing area in the WPF Designer, which offers the WYSIWYG view of the application. It is a place where you can design the outlook of your application, such as drop the controls, change their outlook, size. It also enables designers to zoom in and out and set margins for the controls.
  - XAML view*—Refers to the area in the WPF Designer that allows you to work with the XAML code. The XAML view can also be used to design the UI of WPF applications. When any control is added or modified in the Design view, the corresponding XAML code is automatically added in the XAML view and vice versa.
  - Split view bar*—Refers to a long bar that splits the WPF Designer into the Design view and XAML view. The split view bar appears between the Design and XAML views and separates the Design and XAML views, allowing designers to work with both the views simultaneously. The separation of the WPF Designer into Design view and XAML view is known as the split view. The split view bar has various buttons that assists designers to collapse the Design or XAML view, resize the views, and swap the views.
  - Tag navigator*—Refers to the horizontal bar that appears at the bottom of the WPF Designer and allows designers to navigate to the parent or child XAML elements of the currently selected element.

**12. What is XAML? Explain the use of XAML in WPF applications. Are there any advantages of incorporating XAML in WPF applications?**

■ XAML is a declarative markup language based on XML and introduced by Microsoft Corporation. This markup language uses various predefined markup tags or elements. The tags or elements have attributes that are set using the name-value pairs. Due to its simple and declarative nature, XAML is supported by many technologies, such as WPF, WF, and Silverlight. In WPF, XAML is used primarily to design the UI of the applications. The controls available in WPF are mapped to the XAML elements, while the properties of the controls are mapped to the attributes of the elements. When you add a control to a WPF application, the corresponding XAML element is created automatically. Similarly, when you create an XAML element in a WPF application, the corresponding control is added automatically. The designers create the XAML elements to define the UI of a WPF application. It can use the application developer in the code-behind file to define the logic or functionality of those elements. The developer can also create XAML elements at run time by defining them in the code-behind file.

**13. What is the extension of XAML files?**

■ The XAML files extension is .xaml, for example, Window1.xaml and Page1.xaml.

**14. What are markup extensions and what are its benefits? Write the names of some common markup extensions available in WPF.**

■ A markup extension is a means of setting the value of a property to an object reference rather than a string or numeric value. Markup extensions can be used to set a property value such that the value is based on another element or object. With markup extensions, the assignment of property values can be delayed until run time. Some of the common markup extensions in WPF 4.0 are:

- **Binding** — Allows data-bound values for a property
  - **StaticResource** — Allows an existing WPF resource to be used as the property value
  - **DynamicResource** — Allows a WPF resource to be used as the property value only at run time
  - **TemplateBinding** — Allows the property value in the control template of an element to be used as the property value
- All the markup extensions are objects of the classes that inherit the System.Windows.Markup.MarkupExtension class.

**15. How is the property system of WPF different from the traditional CLR property system? What are dependency properties?**

■ In WPF, properties play an important role in the design and logic of the applications. This is because WPF follows a data-driven approach in which, the primary focus is on the data or values and properties provide an easy and quick way to work with data. The property system of WPF includes the traditional CLR properties as well as dependency properties. A dependency property is a property that depends on various other factors or inputs for its value. The other factors or inputs may include data-bound values, values set through styles, animations, resources, and system-specific or user-specific factors. This implies that the value of a dependency property can be set directly or indirectly through other inputs. Dependency properties are used in the same way as the CLR properties; however, dependency properties are implemented differently. The elements or classes that implement dependency properties are derived classes of the DependencyObject class. Each dependency property is an object of the DependencyProperty class. The dependency property then registers with the WPF property system and wrapped in a CLR property.

**16. What is an attached property? Give an example.**

■ Attached properties are dependency properties of an element that can access and set on any of the child elements of the element. If an element has an attached property, then the attached property is accessible to all the child elements of that element. Most attached properties pertain to the appearance or layout of the elements. For example, the Grid element has the attached properties: Row and Column. The Row property can be used by a child element of the Grid element to indicate the row of the grid in which it appears. The Column property of the Grid element, when used in a child element, indicates the column in which it appears.

The preceding example can be implemented, as shown in the code snippet:

```
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="150" />
        <ColumnDefinition Width="150" />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="50" />
        <RowDefinition Height="50" />
    </Grid.RowDefinitions>
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="50" />
            <ColumnDefinition Width="50" />
        </Grid.ColumnDefinitions>
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="50" />
                <ColumnDefinition Width="50" />
            </Grid.ColumnDefinitions>
            <Text>A</Text>
            <Text>B</Text>
        </Grid>
    </Grid>
</Grid>
```

```

<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="23" />
        <ColumnDefinition Width="17" />
        <ColumnDefinition Width="23" />
        <ColumnDefinition Width="0" />
    </Grid.ColumnDefinitions>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="17" />
            <RowDefinition Height="17" />
        </Grid.RowDefinitions>
        <Button Content="Button1" Grid.Column="0" Grid.Row="0" />
        <Button Content="Button2" Grid.Column="1" Grid.Row="0" />
        <TextBlock Text="TextBlock" Grid.Column="2" Grid.Row="1" />
    </Grid>
</Grid>

```

The attached properties implementation style in WPF is similar to the dependency properties except that attached properties are not wrapped within a CLR property.

#### 17. What are routed events? What is the difference between bubbling and tunneling events?

■ The routed events are a part of the WPF event enhancement system. The events, which traverse through the element tree of a WPF application, are referred by the routed events. These events travel between the original source element and the root element. The routed events can pass event information between source elements and root elements. It depends on the routing strategy, based on which the events are categorized in tunneling and bubbling events. The difference between bubbling and tunneling events is listed in Table 12.1.

Table 12.1: Differences between Tunneling and Bubbling Event:

Tunneling Event	Bubbling Event
Originates on one element and is handled on the root element. The event then traverses through the element tree up to the original source element. The tunneling event traverses from the top or outer element to the child or inner element.	Generates on one element and is handled on the original source element. The event then travels through the element tree up to the root element. The bubbling event traverses from the child element to the root or outer element.

These events are also known as preview events as their names are prefixed with Preview for example PreviewMouseDown and PreviewMouseUp events are tunneling events.

Some examples of bubbling events are MouseDown and KeyUp events.

#### 18. What is a Binary Application Markup Language (BAML) file?

■ BAML files (or Binary XAML files) are compiled XAML files with the .bam extension. These files are included in the assembly that is created in a WPF application at the compile time.

#### 19. Which of the following is the new control introduced in WPF 4.0?

- a. Expander
- b. Grid
- c. Calendar
- d. TextBlock

■ c. The new control introduced in WPF 4.0 is the Calendar control.

#### 20. What are Container controls?

■ The Container controls contain other controls or elements. These controls provide the predefined layout for its sub controls. Some of the common container controls in WPF are:

- *Grid* —Provides a grid with one or more cells that are formed by creating rows and columns. The grid control's every cell can contain sub controls.
- *UniformGrid* —Provides a grid in which the cells are of the same size. The control cells are automatically adjusted to hold the controls.
- *Canvas* —Refers to a built-in layout that enables accurate positioning of its sub controls by using coordinates.
- *DockPanel* —Refers to a built-in layout that enables its sub controls to get arranged in a vertical or horizontal manner along its boundary.
- *StackPanel* —Provides a built-in template that allows its sub controls to be arranged in a vertical or horizontal stack.

#### 21. Define resources in WPF. What are the different types of resources?

■ The resources are objects, elements, or values contained in the resource dictionary. They can be reused in a WPF application. The resource dictionary is an object of the ResourceDictionary class and it contains all the resources defined on an element. You can use an element Resource property to access the resource dictionary. You can define more than one resource of single element. However, every resource must be a unique key. A unique key is a string to identify the resources uniquely in the resource dictionary of that element. The x:Key attribute of the resource specifies the key of the resource. After defining the element resource, typically the root element, the resource works as a reference property to other elements.

Depending upon how WPF resources are referenced, resources are of the following two types:

- *Static resources*—Use StaticResource markup extension to work as a reference in the element that uses the resource. The static

resource values depend at the time of loading and remain static throughout the execution of the application.

- *Dynamic resources*—Use DynamicResource markup extension to work as a temporary expression during the initial compilation in the element that uses the resource. It provides flexibility as it assigns the value at runtime by assigning a property value.

## 22. What is the difference between Simple controls and Content controls?

- The Simple controls cannot hold any content or child control. Some of the examples of Simple controls are TextBox, PasswordBox, and ProgressBar. The Content controls can contain only a piece of content or single child control. The Content property provides the content for these controls. The examples of Content controls are Button, Label, GroupBox, and UserControl.

## 23. What is the function of the BlackOutDates property of the DatePicker control?

- The BlackOutDates property of the DatePicker control retrieves or assigns a collection of dates that are marked as not selectable in the DatePicker control.

## 24. What does a control template do?

- The WPF controls provide the template to define the look, style, and state of the controls. Every WPF control provides the default control template. However, you can create a new template to provide a new look and behavior to a control. The control template is an instance of the ControlTemplate class, which inherits the FrameworkTemplate class.

## 25. What role does a Data template play in WPF controls?

- The Data templates enable you to modify the presentation of the data or content of a WPF control. You can create your own data template and override the default presentation style of the control content. The data template supports two types of controls, content controls and list controls. The data templates are objects of the DataTemplate class.

## 26. What are triggers?

- The triggers are objects that raise an action when a particular condition occurs. Suppose you want to change the background color of a button control when user move the mouse over it then you should use trigger to implement this action. In WPF, triggers are used

to set property values or perform actions, the following are the two types of triggers in WPF:

- *Property triggers*—Sets the property value and raises the action when any changes happen in the property value. When the trigger removes the condition, the changes made by the trigger in property values are undone. Property triggers can perform more than one task through Trigger.EnterActions and the Trigger.ExitActions properties.

- *Event triggers*—Raises an action when a particular event occurs. Once the event has occurred, the task performed by an event trigger cannot be reversed back. Event triggers are instances of the EventTrigger class and are represented by the EventTrigger element.

## 27. What do you understand by WPF commands?

- The commands work as a mediator between the user and the application. Commands and event handlers are different from each other, as commands separate the action from its logic. Due to this, multiple elements can call the same command logic, which can be customized for different targets. A command source calls the command and executes it on a command target. The WPF commands implement the ICommand interface and the command sources implement the ICommandSource interface. If you do not specify the command source, then it uses the keyboard focus to become the command target. For example, if the MenuItem control has TextBox and a PasswordBox controls and it invokes the Paste command without specifying the command target, then the target of the Paste command can be either of the two controls depending on which control has the keyboard focus.

## 28. Which classes are responsible for integrating WPF and Windows Forms technologies?

- The WindowsFormsHost class hosts the Windows Forms control in a WPF application and the ElementHost class hosts the WPF control in a Windows Forms application. Both classes exist in the System.Windows.Forms.Integration namespace.

## 29. Name the classes that allow you to perform transformation for typography and text.

- The classes that pertain to transforming the text in WPF are:
  - The Transform class
  - The TranslateTransform class
  - The RotateTransform class

- The ScaleTransform class
- The SkewTransform class

30. Which namespace allows you to work with documents in WPF?

- The System.Windows.Documents namespace allows you to work with documents in WPF.

31. Explain the types of documents supported by WPF.

■ WPF supports the following two types of documents:

- Fixed documents* —Refers to the documents whose content is static or fixed. In other words, the fixed documents are typeset or ready-to-print documents. The layout of the entire content of static documents remains fix, irrespective of the screen or print device used. The document is printed as it is shown on the screen. Therefore, these are suitable for desktop publishing and word processing applications. It is an object of the FixedDocument class.

- Flow documents* —Refers to the documents that allow flexible and dynamic arrangement of the content; therefore, the document is optimized to view and read the content. It adjusts itself according to the window size, display resolution, or print resolution. There are many options for page viewing, such as single-page, two-page, and continuous scrolling viewing modes. Flow documents also contain a zoom control that facilitates zooming in or out of the document content. It is an object of the FlowDocument class.

32. Write the name of the control that allows you to work with fixed and flow documents.

- The DocumentViewer control allows working with the fixed documents in WPF. Any of the following controls can be used for a flow document:
  - The FlowDocumentScrollViewer control
  - The FlowDocumentReader control
  - The FlowDocumentPageViewer control
  - The RichTextBox control

33. What do you understand by serialization and deserialization of documents?

- While working with WPF documents, the documents are created and stored in a data store of the application. The documents are then

loaded from the data store to the memory. When a document is loaded from the in-memory representation to the data store, it is known as serialization. When the document is loaded from the data store to the memory, then it is known as deserialization. Serialization of documents is performed by calling the write method to save the document, while deserialization of documents is performed by the read method to access the documents from the data store. The System.Windows.Documents.Serialization namespace contains the classes that are used to serialize and deserializing documents.

34. Mention the important classes in WPF that allow you to work with the 2-D graphics.

- The WPF classes that pertain to 2-D graphics are:
  - The *System.Windows.Shapes.Shape class* —Provides the functionality to work with simple 2-D shapes, such as ellipse and rectangle
  - The *System.Windows.Media.Geometry class* —Provides the functionality to work with geometric shapes and curves
  - The *System.Windows.Media.Drawing class* —Provides the functionality to work with 2-D drawings

35. What are the brushes that are available in WPF?

- The brushes available in WPF are objects of the classes that inherit the System.Windows.Media.Brush class. The brushes that are available in WPF 4.0 are:
  - SolidColorBrush* —Paints a given area with a solid color
  - LinearGradientBrush* —Paints a given area with linear gradient
  - RadialGradientBrush* —Paints a given area with radial gradient
  - DrawingBrush* —Paints a given area with a drawing
  - ImageBrush* —Paints a given area with an image
  - VisualBrush* —Paints a given area with a Visual object

36. How many types of projection cameras are available for 3-D graphics in WPF?

- WPF supports mainly two types of projection cameras for 3-D objects:
  - Orthographic camera* —Renders the 3-D scene, such that the exact scale of the object is preserved
  - Perspective camera* —Renders the 3-D scene, such that the objects that are far appear smaller, while the objects that are near appear larger

### 37. What are the different kinds of animation supported in WPF?

- WPF has a rich support for animations, which can be used to perform various tasks, such as change property values, reposition elements, and change colors. The functionality to create and manipulate animations is provided by the `System.Windows.Media.Animation` namespace. Animation in WPF can be of three types:

- **Basic animation**—Refers to the animation in WPF that is created by changing the property value from a starting value to a destination value. Alternatively, an offset value can also be provided to indicate the amount of change during the animation. The starting value is specified by the `From` property, the destination value is specified by the `To` property, and the offset value is specified by the `By` property. This is why basic animation is also known as From/To/By animation.

- **Keyframe animation**—Refers to the animation that is created by using keyframes, which are the destination values for a property. There can be several keyframes in a single keyframe animation.

- **Path animation**—Refers to the animation that is created by changing the property value with respect to a given path described by a `PathGeometry` object. It is ideal for situations when the movement of an object or element has to be along a path. The lines and curves in the given path determine the property value.

**Abstract Class**—Refers to the class that is the base class for basic animation ands keyframe animation.

### 38. What are the image formats supported in WPF?

- The file formats of images that are supported in WPF are:
  - Bitmap (.bmp)
  - Joint Photographic Experts Group (.jpg and .jpeg)
  - Portable Network Graphics (.png)
  - Tagged Image File Format (.tiff)
  - Graphics Interchange Format (.gif)
  - Icon (.ico)
  - Windows Media Photo

### 39. What are the formats of audio-video data that WPF supports?

- WPF supports the Windows Media Video (.wmv), Advanced Systems Format (.asf), Moving Picture Expert Group (.mpeg) and Audio Video Interleave (.avi) file formats for audio-video data.

### 40. What are the two modes of media playback?

- The `MediaPlayer` and `MediaElement` controls have two modes in which media can be played back. These two modes depend on the `Clock` property of the controls. The two playback modes are:
  - **Clock mode**—Refers to the mode when the `Clock` property is set to any value other than the null value. The methods of the `ClockController` object are used to control the playback of the media.
  - **Independent mode**—Refers to the mode when the `Clock` property is set to the null value. The `Play`, `Pause`, `Close`, and `Stop` methods can be used to control the playback of the media in the independent mode.

### 41. What is the technique that is used by WPF applications to provide Help to the users? How it is different to the Help, which Windows controls provide to their users through pop-ups or online help?

- The `HelpProvider` control offers help to the users of Windows Forms through pop-ups and online help. The `HelpProvider` control is registered in the `system.windows.forms.dll` assembly. However, WPF does not have any control which is equivalent to the `HelpProvider` control. WPF applications do not offer any Help when users press the **F1** key. Instead, Users get Help through ToolTips and Enhanced Tool Tips. The tool tips show images, icons, and other previews.

### 42. What is the current version of Silverlight, which is supported by .NET Framework 4.0?

- The current version of Silverlight, which is supported by .NET Framework, is 4.0.

### 43. What is Silverlight?

- Silverlight is a powerful environment to develop an interactive application for the Windows, Web, and mobile devices. It is a cross-browser, cross-platform, and cross-device environment. Both Microsoft Windows and Apple Macintosh family of operating systems as well as popular Web browsers, such as Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari, support Silverlight.

### 44. Write the names of Web browsers that support Silverlight.

- The Web browsers that support Silverlight are Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari.

**45. What are the tools or software packages available in the market that you can use to work with Silverlight?**

- The three primary software packages that can be used to work with Silverlight are Visual Studio 2010, Expression Studio, and ComponentOne Studio.

**46. What is an application package in Silverlight?**

- An application package in Silverlight is a file with the .xap extension. When Silverlight applications compile, the Application packages are created at that time. It is the unit of deployment to Silverlight application. It consists of the necessary assembly files and manifests to run the application.

**47. What are the primary languages that Silverlight uses?**

- Silverlight uses the following languages:
  - Hypertext Markup Language (HTML)
  - JavaScript
  - Extensible Application Markup Language (XAML)

**48. Define the Deep Zoom technology in Silverlight.**

- The Deep Zoom technology in Silverlight 4.0 facilitates the zooming in and out of high-resolution images. The Deep Zoom technology provides developers so many functionalities to make a creative and uniquely interactive Web application. It allows users to zoom in or out of multiple photographs. The photographs provide a highly detailed and fine view.

**49. What are the networking services that are supported by Silverlight?**

- The various networking services that Silverlight supports are:
  - Hypertext Transfer Protocol (HTTP)
  - Really Simple Syndication (RSS)
  - Representative State Transfer (REST) services

**50. What is a test page in Silverlight and when is it created?**

- A test page is an HTML page that hosts the Silverlight plug-in in an <object> tag. It appears when users use a Silverlight application. It is created when a Silverlight application is compiled.

**51. Write the name of the control that can be used to enable the users to adjust the space in a grid.**

- The GridSplitter control in Silverlight 4.0 enables users of the Silverlight application to redistribute the space between the rows and columns of the grid.

**52. Is it possible to work without adding the ScriptManager control in a Silverlight-enabled ASP.NET Web page?**

- No, you cannot work without the ScriptManager control in a Silverlight-enabled ASP.NET page. This control handles the Silverlight content in the Web page.

**53. What is the use of the MediaPlayer server control in ASP.NET?**

- The MediaPlayer server control in ASP.NET 4.0 provides a media player to playback media in Web pages. You can include audio and video in your Web application by the MediaPlayer control. You can also change the skin of the MediaPlayer control.

**54. What are the file formats supported by the MediaPlayer server control in ASP.NET?**

- The MediaPlayer server control supports the Windows Media Audio (WMA) and MP3 files for audio and the Windows Media Video (WMV) files for video.

**55. What is the functionality provided by the Silverlight server control in ASP.NET?**

- The Silverlight control allows the developers to display any XAML content. In this control, managed assemblies, application packages, and client-side JavaScript libraries are embedded.

**56. Demonstrate how the Popup control in Silverlight is used with the help of an example.**

- The Popup control provides the description to other controls, such as a button or checkbox. It appears only for a single UI control. It means to provide description for multiple controls that need to be encapsulated within a container control. The Popup control can be used as shown in the following XAML code snippet:

```
<Popup Isopen="True" HorizontalOffset="20"
VerticalOffset="50" >
<Popup.Child>
<Button x:Name="button1" Content="Hello"
Height="30" Width="100"/>
</Popup.Child>
</Popup>
```

## 57. What is the difference between Silverlight applications and XBAP applications in WPF?

- The differences between Silverlight and XBAP applications are listed in Table 12.2.

Table 12.2: Differences between XBAP and Silverlight Applications

WPF Applications	Silverlight Applications
XBAPs can run only in Internet Explorer and Firefox Web browsers.	Silverlight provides the necessary framework components through its .NET Framework Silverlight component. XBAP has access to the complete WPF functionality.

## Windows Communication Foundation 4.0

### 1. What is Windows Communication Foundation (WCF)?

- WCF is a distributed technology that provides a single, integrated platform or model called the service model to develop distributed applications for Windows. With WCF, powerful and interoperable service-oriented distributed applications can be developed. WCF was introduced in .NET Framework to resolve the problem of inconsistency in the communication between different applications due to the usage of diverse distributed technologies, such as Microsoft Message Queuing (MSMQ), .NET remoting, XML Web Services, and Enterprises Services (COM+ services). WCF not only resolves the inconsistency but also ensures the security and reliability of the distributed application and interoperability among the distributed technologies. Similar to the other distributed technologies, WCF also uses the Simple Object Access Protocol (SOAP) messages to exchange information between the service and the clients.

### 2. What are the benefits that WCF offers?

- The major benefits that WCF offers include:
  - *Productivity*—WCF is one of the fastest distributed technologies available today facilitating better performance and higher efficiency of the applications.
  - *Interoperability*—WCF supports interoperability between different platforms that use SOAP as well as interoperability with previous distributed technologies, such as MSMQ, remoting, and Web services.
  - *Security*—WCF has several security measures, such as service endpoint authentication, client principal authentication, and message integrity and confidentiality. It also has support for Windows integrated security and transport security measures, such as HTTPS, and message security measures, such as WS-Security.

# 13

- Reliability**—WCF has support for reliable sessions between the service, clients, and queues for separation between the service and clients. Reliable sessions make use of the WS-ReliableMessaging protocol and ensure that the SOAP messages sent from one endpoint to another are received only once and in the same order. Queues in WCF are based on MSMQ and allows loosely coupled applications, load leveling, and disconnected operations.

- Support for transactions**—WCF supports transactions, which refers to a single executable unit consisting of different operations grouped together. The WCF transactions are characterized by atomicity, consistency, isolation, and durability. It also has support for the WS-AtomicTransaction protocol to facilitate the use of transactions by interoperable applications.
- Support for AJAX and JSON**—WCF enables services to expose their operations to AJAX clients through the AJAX and JSON data transfer formats.

### 3. What was the code name of WCF?

- The code name of WCF was Indigo.

### 4. When was WCF introduced?

- WCF was introduced with .NET Framework 3.0 in 2006.

### 5. Mention the different layers of WCF.

- WCF has serviced-oriented architecture and supports services as well as clients. A WCF application can act as both service and client. The WCF services and clients use a message-based system, that is, the services and clients communicate with one another through messages. WCF provides different layers for this message system. There are four layers in WCF, which are as follows:

- Contracts**—Defines the different functionalities that are exposed by the service to the client
- Service runtime**—Defines the various run time behaviors of the service

- Messaging**—Defines the channels that are used for processing and exchanging messages

- Activation and Hosting**—Defines the way a WCF service is hosted

### 6. What is an endpoint? Briefly explain the components of a WCF endpoint.

- In WCF, services and clients communicate through endpoints. An endpoint is the point where the services and clients meet and exchange messages. The services expose their functionalities to the client through their endpoints. A single WCF service can have one or more endpoints. An endpoint comprises the following components:
  - Endpoint address**—Refers to the address of the service, that is, where the service can be found. An endpoint address needs to be unique and is an object of the EndpointAddress class. The endpoint address needs to be a Uniform Resource Identifier (URI) address and needs to include an Identity object that represents the security identity of the service.
  - Binding**—Refers to the means in which a client interacts with the endpoint. Binding encompasses the transport protocol and the encoding scheme used for communicating between the service and client.
  - Service contract**—Refers to the set of functionalities and operations exposed by the service to the clients.
  - Behavior**—Refers to the local implementation or behavior of the endpoint. For example, the endpoint can behave in such a way that it listens to the requests from clients at a particular URI address.

### 7. Discuss the bindings that are available for WCF applications.

- In WCF, bindings refer to objects that define how to connect and communicate with a service endpoint. A binding can specify the transport protocol, encoding scheme, and security mechanism required to communicate with the service endpoint. In WCF, there are several system-provided bindings that include a predefined combination of the transport protocols, encoding schemes, and security measures. These system-provided bindings are suitable for most of the WCF services and client. Some of the important system-provided bindings are:
  - BasicHttpBinding**—Refers to the binding that uses the HTTP protocol, text/XML message encoding, and transport security. This binding is ideal for Web services that conform to the WS-Basic Profile specification.
  - WSHttpBinding**—Refers to the binding that uses the HTTP protocol and the WS- specification security mechanism. This binding is ideal when support for transactions and WS-addressing is required.

- *NetNamedPipeBinding* —Refers to the binding that uses the Named Pipes protocol, binary message encoding, and transport security. This binding is suitable when the service and client are on the same machine.
  - *NetMsmqBinding* —Refers to the binding that uses the MSMQ protocol and is ideal when the service and client are on different machines.
- 8. What are the different types of contracts available in WCF?**
- Contracts are objects that specify the functionality of a WCF service that is available to the clients. A contract contains information about the operations available to a client, the format of the message exchanged between the service and client, the kind of response message sent to the client, and the data or input parameters required from the client to perform the operations. WCF has four main types of contracts, which are as follows:
  - *Service contract* —Defines the entire functionality of a WCF service that is exposed to the clients. A service contract groups various operations through which the functionality of the service is exposed into a single entity. To create a service contract, an interface marked with the `ServiceContract` attribute is created.
  - *Operation contract* —Defines the input parameters and the return type of a particular operation in the service. The operation contract can specify information on whether the operation takes and returns a single message or takes one or more values and returns a single value. Each operation contract is specified by a method marked with the `OperationContract` attribute. There can be multiple operation contracts in a single service contract.
  - *Data contract* —Defines the data that is exchanged between the operations of the service. A data contract specifies the data types that are used in the service. Data contracts are beneficial when there are complex data types; for simple data types, data contracts are not required. To create a data contract, a class marked with the `DataContract` attribute is created along with its members marked with the `DataMember` attribute.

- 9. Briefly explain the basic steps required to create a service contract in WCF.**
- WCF services are implemented as classes. Each WCF service is characterized by a service contract. Essentially, a service contract contains operations, which are accessed by the clients. You need to perform the following steps to create a service contract:
    1. Create an interface that signifies the service contract by marking the interface with the `ServiceContract` attribute.
    2. Declare methods that represent the various operations of the service. These methods need to be marked with the `OperationContract` attribute.
    3. Create a class to implement the interface that represents the service contract.
    4. Define the methods that are declared in the service contract interface.

**10. Discuss the message patterns supported by a service contract in WCF.**

    - WCF services communicate with clients through messages. The messages are generated by the operations defined in the service. Depending on the signature of the operation, messages have a pattern, which indicates how the messages can be exchanged between a client and a service. In WCF, there are three message patterns:
    - *Request/Reply pattern* —Refers to the message pattern wherein the client gets a reply with respect to a request that it sends to the service. This message pattern is a two-way operation where each request by the client has a reply from the service. The Request/Reply message pattern is the default message pattern. It is characterized by an operation with one or more parameters and a return value.
    - *One-way pattern* —Refers to the message pattern wherein the client does not wait for a reply from the service for further processing. It is a one-way operation and is analogous to the fire-and-forget style of messaging. It is similar to an asynchronous method called with a void return type. To specify a one-way pattern for an operation, the return type of the

operation has to be void and the `IsOneWay` property of the corresponding operation contract has to be true.

- Duplex pattern**—Refers to the message pattern wherein the service and client can send messages to each other such that a peer-to-peer communication pattern is established. The service and client can simultaneously act as the sender and the receiver.

#### 11. Explain the instancing model of WCF.

■ WCF consists of an instancing model that specifies how many instances of the WCF service are created and run at a particular time. Depending upon the number of instances created by the service, there are three types of services:

- Per-Call service**—Refers to the service that is created for every operation requested by a client, that is, every time a client requests for an operation, a new instance of the service is created. The per-call service can be called only when the client call service, the `InstanceContextMode` property of the `ServiceBehavior` attribute has to be set to `PerCall`.

**Per-Session service**—Refers to the WCF service that is created for each session between a client and the service. The instance of a per-session service does not depend on other instances of the service and exists as long as the client needs it. Calls to the service in the same session are handled by the already created instance. To specify a WCF service to behave as a per-session service, the `InstanceContextMode` property of the `ServiceBehavior` attribute has to be set to `PerSession`.

**Singleton service**—Refers to the WCF service that is created for the entire host application, that is, only one instance of the service is created for the entire lifetime of the application. This single instance of the service handles all the requests for the clients. This implies that the service can handle only one message at a time. When the host application ends, the instance of the singleton service is also destroyed. To configure a service as singleton, set the `InstanceContextMode` property of the `ServiceBehavior` attribute to `Single`.

#### 12. What does the **Svcutil.exe** tool perform?

- The **Svcutil.exe** tool generates a client-side proxy (a language-specific file, for example a `.cs` or `.vb` file) and configuration file (`.config` file) from the service metadata.

#### 13. Which protocol does WCF employ for distributed applications?

- WCF applications employ the Simple Object Access Protocol (SOAP).

#### 14. What do you understand by service identity?

- WCF service identity is a value generated from the WSDL for the service at design time and sent to a client, which it uses to authenticate the service before requesting any operation of the service. The service identity is represented by the `Identity` property of the `EndpointAddress` class. The **ServiceModel Metadata Utility Tool** (**Svcutil.exe**) allows you to access the service identity of a service. This tool generates a configuration file for the WCF service containing the service identity.

#### 15. What do you understand by hosting a WCF service and what are the different ways in which you can host a WCF service?

- WCF services need to be hosted before they can be used by clients. This is because a WCF service is basically a class that implements the service and is compiled into a library, which needs to be hosted inside another application. WCF provides the following four ways in which WCF services can be hosted:
  - Self/hosting**—Specifies that the services are self-hosted in a Graphical User Interface (GUI) application such as Console, Windows Forms, or WPF application. You need to write the code to initialize the hosting environment. WCF provides you the `ServiceHost` class for self-hosting the services.
  - Windows Service**—Specifies that the WCF service is hosted as a Windows Service that starts when a system boots. To enable this hosting option, you need to register the application domain, which hosts the WCF service as Window Service so that the Service Control Manager (SCM) controls the hosted WCF service.
  - IIS Server**—Specifies that the WCF service is hosted in the IIS server and is automatically incorporated into ASP.NET. As a result, the WCF service can benefit from the inherent features of ASP.NET, such as process recycling, idle-time management, and activation. Such a hosting technique does not require any additional hosting code. The WCF service can be hosted in IIS by using the traditional Web hosting technique used by Web applications and Web services.

- **Windows Activation Service (WAS)**—Specifies a new activation mechanism to retain the process model of IIS and hosting features. WAS automatically activates WCF applications when they are deployed on a computer that is running WAS. Other than Hyper Text Transfer Protocol (HTTP), WAS also supports other protocols, such as Transmission Control Protocol (TCP) and Named Pipes.

#### 16. What are the new features in WCF 4.0?

- WCF 4.0 has introduced various new features to increase the productivity of developers working with WCF services. These new features of WCF allow you to implement more sophisticated service-oriented patterns in building smarter and more resilient services. For example, in WCF 4.0, you do not need to manually configure the endpoints, behaviors, and other settings, as these are automatically configured by WCF. The new features of WCF 4.0 are listed as follows:
  - **Routing Service**—Acts as a broker or gateway to the services and can be implemented as a centralized security boundary in applications so that all the incoming messages can pass through it. The purpose of the routing service is to handle the messages over the WCF-supported protocol. WCF 4.0 provides the `RoutingService` class to implement routing services in applications. The `RoutingService` class receives incoming messages and sends them to the target service.
  - **Default configuration**—Specifies that in WCF 4.0, you never need to have adequate knowledge of the configuration of WCF to define an endpoint, behavior, and other settings of WCF, as required in the previous versions of WCF. In WCF 4.0, you do not need to manually configure the endpoints, behaviors, and other settings, as these are automatically configured by WCF. These configurations are added to the `Web.config` file of the application while creating the WCF service application. You can also change the default configuration in the application according to your requirements.
  - **Support for discovery behaviors**—Specifies that WS-Discovery is a technical specification, which is used to dynamically discover the location of the WCF service endpoints by defining SOAP-based protocol. SOAP allows clients to examine the WCF service endpoints based on certain criteria to retrieve a list of the suitable candidates offering WCF services. The client can then select an endpoint from the discovered list and use it as a current runtime location of the WCF service endpoints.

#### 17. What are the features of WCF WebHttp Service?

- The WCF **WebHttp** service includes the following features:
  - **Support for automatic help page**—Helps a consumer to understand the RESTful web service
  - **Simplified HyperText Transfer Protocol (HTTP) caching**—Allows you to cache the responses of messages from WCF Web HTTP service operations
  - **Message format selection**—Allows services to handle or return the messages in either the XML or JSON format
  - **REST-friendly exceptions**—Transmits HTTP errors back to consumers by using a new class, `WebFaultException<T>`
  - **New Visual Studio project templates**—Allows you to implement new project, control, and tools into your Visual Studio 2010 applications by using Extension Manager.

#### 18. What is throttling in WCF 4.0?

- WCF throttling provides you with some properties that help you to limit the number of instances and sessions that you can create at application level.

- **Improved REST service development**—Provides new features for building the **RESTful web services** (now referred as WCF **WebHttp** services) using WCF. A RESTful web service is a service based on HTTP protocol and REST principles. The REST principles are defined by three categories with respect to a Web service. As per the REST principles, a service:
  - Can be accessed with a simple URI
  - Supports MIME types
  - Uses different HTTP methods

- The three WCF throttling settings are:
- MaxConcurrentCalls*—Limits the total number of concurrent calls that you can process simultaneously across all service instances
  - MaxConcurrentSessions*—Limits the total number of sessions that a single ServiceHost object can accept
  - MaxConcurrentInstances*—Limits the total numbers of service instances that a serviceHost object can create at one time

# 14

## ASP.NET AJAX

### 1. What is ASP.NET AJAX?

- ASP.NET AJAX, mostly called **AJAX**, is a set of extensions of ASP.NET. It is developed by Microsoft to implement **AJAX** functionalities in Web applications. ASP.NET AJAX provides a set of components that enable the developers to develop applications that can update only a specified portion of data without refreshing the entire page. The ASP.NET AJAX works with the **AJAX Library** that uses object-oriented programming (OOP) to develop rich Web applications that communicate with the server using asynchronous postback.

### 2. What do you mean by the term asynchronous postback?

- The ASP.NET AJAX allows developers to use the asynchronous postback when a browser sends the callback to a server by establishing a new connection; thereby, making the callback out-of-band. Due to this, the entire page is not submitted to the server. Moreover, if the client has sent the asynchronous request to the server, he can continue his work while the server is processing the request. Figure 14.1 illustrates how the server responds according to the client request without refreshing the entire page:

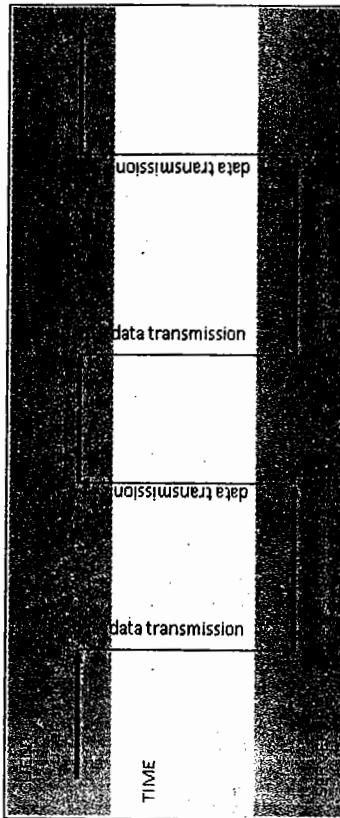


Figure 14.1: Displaying asynchronous postback.

### 3. What is the difference between synchronous postback and asynchronous postback?

- The difference between synchronous and asynchronous postback is as follows:

- Asynchronous postback renders only the required part of the page; whereas, synchronous postback renders the entire page for any postback
- Asynchronous postback executes only one postback at a time, that is, if you have two buttons doing asynchronous postback, the actions will be performed one by one; whereas, synchronous postback executes all the actions at once
- Asynchronous postback only modifies the update panel that raises the postback; whereas, synchronous postback modifies the entire page

### 4. What technologies are being used in AJAX?

- AJAX uses four technologies, which are as follows:

- JavaScript
- XMLHttpRequest
- Document Object Model (DOM)
- Extensible HTML (XHTML) and Cascading Style Sheets (CSS)

### 5. Why do we use the XMLHttpRequest object in AJAX?

- The XMLHttpRequest object is used by JavaScript to transfer XML and other text data between client and server. The XMLHttpRequest object allows a client-side script to perform an HTTP request. AJAX applications use the XMLHttpRequest object so that the browser can communicate to the server without requiring a postback of the entire page. In earlier versions of Internet Explorer, MSXML ActiveX component is liable to provide this functionality; whereas, Internet Explorer 7 and other browsers, such as Mozilla Firefox, XMLHttpRequest is not liable to.

### 6. How can we get the state of the requested process?

- XMLHttpRequest get the current state of the request operation by using the readyState property. This property checks the state of the object to determine if any action should be taken. The readyState property uses numeric values to represent the state.

### 7. List the different states of XMLHttpRequest with their description.

- The different states of the XMLHttpRequest object are as follows:

- Uninitialized*—Refers to the state when the object has not been initialized

- Open*—Refers to the state when the object has been created; however, the send function has not been invoked

- Sent*—Refers to the state when the send function is invoked; however, the status and headers are not available

- Receiving*—Refers to the state when the process is receiving data

- Loaded*—Refers to the state when the procedure is completed and the entire data is available

### 8. What are the different ways to pass parameters to the server?

- We can pass parameters to the server using either the GET or POST method. The following code snippets show the example of both the methods:

Get:

```
XMLHttpRequest.open("GET", "test.aspx?value", true);
```

Post:

```
XMLHttpRequest.open("POST", "test.aspx", true);
```

### 9. Is there any difference between HTML and XHTML?

- Extensible HTML (XHTML) is a markup language that provides the mixture expressions of HTML and XML. XHTML is a flexible markup language that enables automated processing by standard XML tools, which was difficult in HTML.

### 10. Briefly describe ASP.NET AJAX Framework.

- ASP.NET AJAX Framework provides a platform where developers can develop such type of applications that use the AJAX concept. The AJAX provides the collection of technologies to create dynamic pages at the client side. The JavaScript requests are responsible to retrieve data from the server or send data to the server. Even some processing at server also requires handling requests, such as searching and storing of data. These tasks are achieved more easily using the AJAX Framework.
- AJAX Framework is completely devoted to process requests. The objective of the AJAX engine is to reduce the delays that the user notices while performing a postback to the server. AJAX Framework allows JavaScript functions to send requests to server at the client side. On the other side, it allows the server to process the client's request, searches data, and responds the result to the browser.

**11. Explain the limitations of AJAX.**

- The following are the limitations of AJAX:
  - It is difficult to bookmark a particular state of the application
  - Function provided in the code-behind file do not work because the dynamic pages cannot register themselves on browsers history engine automatically
  - If JavaScript is disabled, then AJAX is not able to perform any work
  - Response time may be slow because different controls of a page are loaded at different time

**12. What are the differences between AJAX and JavaScript?**

- The differences between AJAX and JavaScript are given as follows:
  - AJAX sends request to the server and does not wait for the response. It performs other operations on the page during that time. JavaScript make a request to the server and waits for response.
  - AJAX does not require the page to refresh for downloading the whole page while JavaScript manages and controls a Web page after being downloaded
  - AJAX minimizes the overload on the server since the script needs to request once while JavaScript posts a request that updates the script every time

**13. What is the importance of client-side libraries?**

- Client-side libraries contain built-in code to make asynchronous calls over XMLHttpRequest. These libraries automatically handle browser compatibility issues. These libraries are based on a programming model similar to ASP.NET.

**14. What are the components of the ASP.NET AJAX architecture?**

- You can divide the ASP.NET AJAX architecture into two components— AJAX client architecture and AJAX server architecture.

**15. What are the different controls of ASP.NET AJAX?**

- ASP.NET AJAX includes the following controls:
  - ScriptManager
  - ScriptManagerProxy
  - UpdatePanel
  - UpdateProgress
  - Timer

**16. What is the use of the ScriptManager control in AJAX?**

- The ScriptManager control is a core control that performs a key role in implementing the ASP.NET AJAX functionality. It helps to use JavaScript for the Microsoft AJAX Library. It should be noted that AJAX Library on a Web page can only be used if the Web page contains the ScriptManager control. This control makes use of the ScriptManager class to maintain the AJAX script libraries and script files. It allows for partial page rendering, Web service calls, and use of ASP.NET AJAX Client Library by rendering the AJAX Library scripts to the browser.

**17. Is it possible to use multiple ScriptManager controls on a Web page?**

- No, it is not possible.

**18. What is the role of the ScriptManagerProxy control?**

- A Web page cannot contain more than one ScriptManager control. You can use the ScriptManagerProxy control to add scripts to other pages; however to perform such an operation, you need to work with a master page that contains the ScriptManager control. If you have only few pages that need to register to a script or a Web service, then you should remove these pages from the ScriptManager control and add them as individual pages by using the ScriptManagerProxy control. If you include the scripts on the master page by the ScriptManager control, then the items get downloaded on each page that extends the master page, even if they are not necessary.

**19. Describe the situations in which AJAX should not be used.**

- You should not use AJAX if:
  - You want the page to show in a search engine, such as Google, because WebCrawler does not execute JavaScript code
  - The browser does not support JavaScript
  - You want to create a secure application

**20. How can you find out that an AJAX request has been completed?**

- You can find out that an AJAX request has been completed by using the readyState property. If the value of this property equals to four, it means that the request has been completed and the data is available.

## 21. What is the syntax to create AJAX objects?

- `var myObject = new AjaxObject({pagePath});`

The page path is the URL of the Web page containing the object that you want to call. The URL must be of the same domain as the Web page.

## 22. What is JSON?

- JSON is an abbreviation of JavaScript Object Notation. It is a safe and reliable data interchange format in JavaScript, which is easy to understand not only for the users but also for the machines.

## 23. Explain the UpdatePanel control.

- The UpdatePanel control specifies the portions of a Web page that can be updated together. As the UpdatePanel control refreshes only a selected part of the Web page instead of refreshing the entire page with a postback, you get more flexibility to create rich and client-centric Web applications. Refreshing a selected part of the Web page is referred as partial-page update. You can add one or more UpdatePanel control in the Web page, which automatically participates in partial-page update without custom client script. The UpdatePanel control uses the UpdatePanel class to support the partial-page rendering.

## 24. Why do we use the UpdateProgress control in AJAX?

- The UpdateProgress control is somewhat related to the UpdatePanel control. The UpdateProgress control enables you to design a user-friendly interface when a Web page consists of a number of UpdatePanel controls for partial-page rendering. The UpdateProgress control makes you aware of the status information about the partial-page updates in the UpdatePanel control.

## 25. Explain the need of the Timer control in AJAX.

- The Timer control is used with an UpdatePanel control to allow partial-page updates at a specified interval. It is mostly used when a periodically partial-page update for one or more UpdatePanel controls is required without refreshing the entire page.
- The Timer control is a server control that sets a JavaScript component in the Web page. The Interval property of the Timer control specifies time in milliseconds. Similar to the UpdatePanel control, the Timer control also requires an instance of the

ScriptManager control in the Web page. When the Timer control initiates a postback, the Tick event is raised on the server for which you can provide an event handler to perform the actions when the page is submitted to the server. The Tick event occurs when the time specified in the Interval property has elapsed and the page is posted on the server. You can add one or more Timer controls on a Web page. Usually the entire page requires only a single Timer control; however, you can use multiple Timer controls, if the UpdatePanel controls are being updated at different intervals.

## 26. Can we call server-side code from JavaScript?

- Yes, page methods and Web services are the two techniques to call the server-side code from JavaScript.

## 27. Can we nest the UpdatePanel controls?

- Yes, we can nest the UpdatePanel control.

## 28. Describe AJAX Control Extender Toolkit

- AJAX Control Toolkit is a set of extenders that are used to extend the functionalities of the ASP.NET controls. The extenders use a block of JavaScript code to add new and enhanced capabilities to the ASP.NET controls. AJAX Control Toolkit is a free download available on the Microsoft site. You need to install this toolkit on your system before using extenders.

## 29. Is the AjaxControlToolkit.dll file installed in the Global Assembly Cache?

- No, you have to copy this file to the Bin folder of your application.

## 30. What are the requirements to run ASP.NET AJAX applications on a server?

- AJAX is a built-in functionality of .NET Framework 4.0. Therefore, you can run an AJAX application by just installing Microsoft Visual Studio 2010. However, to use extenders in your applications, you are required to install AJAX Control Toolkit and copy the AjaxControlToolkit.dll file to the Bin directory of your application.

## 31. What are the new features of ASP.NET AJAX 4.0?

- ASP.NET 4.0 AJAX includes several new features that provide more functionality to a user. These features are as follows:
  - Support for live data binding
  - Support for client-side template rendering

- Support for declarative instantiation of client components
- Support for using the observer pattern on JavaScript objects and arrays
- Support for invoking ADO.NET data services and data contexts
- Support for the DataView control

**32. Explain the live data-binding feature of ASP.NET AJAX 4.0.**

- Live data binding ensures that the data is bound with the server controls at runtime. ASP.NET AJAX 4.0 provides the support for live data binding with the help of observer pattern. The Sys.Observer interface is used to support live data binding. The live data binding can be used in different ways, one-way and two-way. In one-way data binding, the data is not updated automatically, if there is any change in the data source. On the other hand, in the two-way data binding, the data is updated automatically if the data source changes. The code snippet for one-way data binding is as follows:

```
<h3>[L:ProductName]1</h3>
```

The code snippet for two-way data binding is as follows:

```
<input type="text" value="binding ProductName" />
```

**33. What are the new features included in the Microsoft AJAX library?**

- The Microsoft AJAX library is a client-based JavaScript library that is compatible with all modern browsers and offers a lot of functionality as compared to JavaScript. This library is released with new features and fully supports ASP.NET 4.0. The new features included in the Microsoft AJAX library are as follows:
  - *Imperative syntax* —Supports simple imperative syntax that is used to create and manage controls
  - *Script loader* —Retrieves all scripts that are needed by one or more client component or control automatically and executes the scripts in the order in which they are received
  - *Client data access* —Supports to access client data and display by client data control and client template
  - *Client datacontext* —Supports read and write permission to data from a database
  - *The AdoNetDataContext class* —Enables you to easily interact with an ADO.NET Data Services service
  - *Jquery integration* —Helps to access the elements in your Web pages, work with client-side events, enable visual effects, and make it easier to use AJAX in your applications

**34. What are the extender controls?**

- The extender controls uses a block of JavaScript code to add new and enhanced capabilities to ASP.NET. The developers can use a set of sample extender controls through a separate download—AJAX Control Toolkit (ACT).

**35. What are the new controls introduced in ASP.NET AJAX Control Toolkit?**

- The following controls are introduced with the new version of AJAX Control Toolkit:
  - *SeaDragonExtender control* —Refers to the control that is used to deeply zoom the images. You can zoom in or out the image or the particular portion of the image by using the mouse. You can also create a menu over the SeaDragon control. This control is helpful when you want to analyze the image closely.
  - *AsyncFileUploadExtender control* —Refers to the control that provides the facility to upload and save the files on the server asynchronously. You can check the outcome either at the server or client side.

**36. How many validation controls are available in ASP.NET AJAX 4.0?**

- The following validation controls are available in ASP.NET AJAX 4.0:
  - *FilteredTextBoxExtender* —Enables you to apply filtering to a text box
  - *MaskedEditExtender and MaskedEditValidator* —Restricts a user to enter only a certain pattern of characters in the TextBox by applying a mask to the input
  - *ValidatorCalloutExtender* —Attaches to the ASP.NET validators so that the error messages are not displayed as a simple text but as a balloon-style ToolTip
  - *NoBot* —Prevents the spam/bot from filling the input forms automatically and uses the Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA), which is a type of challenge-response test to ensure that the response is not generated by the computer
  - *PasswordStrengthExtender* —Measures the strength of the password text entered within the text box by validating with the different strength specified parameters

### 37. Describe the AccordionExtender control.

- The AccordionExtender control is similar to the CollapsiblePanelExtender control. It allows you to group multiple collapsible panels in a single control. At the same time, it also manages the collapsed and expanded state of each panel; therefore, expanding one panel at a time. In other words, the AccordionExtender control does not support expanding two or more panels simultaneously. Instead, the header templates of all the panels are always visible so that you can click on any of them to display the hidden contents. By default, the AccordionExtender control opens with one panel as expanded.

### 38. What is the work of the ConfirmOnFormSubmit property in the ConfirmButtonExtender control?

- The ConfirmOnFormSubmit property determines whether or not the confirm dialog box should wait when the form is submitted for display.

### 39. What does the DynamicPopulateExtender control do?

- The DynamicPopulateExtender control populates the contents of a control dynamically. It enables you to send an asynchronous call to the server that dynamically populates the contents of a control. The DynamicPopulateExtender control replaces the contents of a control with the result of a Web service or page method call.

### 40. What does the MinimumPrefixLength property of the AutoCompleteExtender control do?

- The MinimumPrefixLength property sets the minimum number of characters that must be entered before getting suggestions from the Web service.

### 41. Explain the Step property of the NumericUpDownExtender control.

- The Step property sets the steps for numeric increment and decrement. The default value is 1.

### 5. Why do you use Windows Installer?

- The Windows Installer deployment technique allows you to deploy Windows-based and Web applications by creating a Windows Installer

**15**

### Deployment

#### 1. What is deployment?

- Deployment refers to the distribution of an application among various end-users. It is a process that makes software available for use by just installing it on the client computer.

#### 2. List different ways of deployment that are supported by .NET Framework 4.0.

- Windows Installer
- ClickOnce
- XCOPY
- Copy Web Site
- Publish Web Site tool

#### 3. What is XCOPY?

- XCOPY enables you to deploy an application by copying the application directory and all subdirectories to the target computer and then executing the application on the client. The application starts executing on the target computer by using its assembly file, which is a self-description file that contains all the information about the application. The XCOPY deployment does not make any impact on the target system while configuring the components and registering entries, and is therefore known as zero-impact installation.

#### 4. Does XCOPY copy the hidden and system files?

- No. By default, XCOPY excludes the hidden and system files. However, you can include the hidden and system files using the /h switch.

Package. The installer package has an extension of **.msi** and it contains the application, any dependent files, registry entries, and the rest. The installer package can then be distributed to various end-users by simply copying it on the target computers. The end-users can then run the installer package to install the application anywhere in their computers. The installation takes place using the installation wizard; therefore, the users can easily install the application on their system. Once your application is installed on the target computer, end-users can open the application from the installed location.

#### 6. What is ClickOnce?

■ ClickOnce is a deployment technology that allows you to publish Windows-based applications on a Web server or a network file share location. Visual Studio 2010 provides the full support for the ClickOnce deployment technology to publish and update the applications. ClickOnce deployment is simpler and easier for both the end-users and the developers. To deploy an application using the ClickOnce deployment technique, you need to publish the application. The ClickOnce deployment allows you to deploy self-updated Windows and Console applications that can be installed, updated, and run from a Web site. It also supports the deployment of WPF XAML Browser Application (XBAPs). The ClickOnce deployment technique allows you to deploy applications from the Web, network file share, or removable media, such as CD-ROM or DVD-ROM. Based on the type and requirements of your application, the ClickOnce deployment technique provides different settings to choose the way you want to deploy your application.

You can choose from any of the following three settings:

**To start from the Web or a network file share**—Deploys an application either on the Web or a shared network. You should apply this setting for applications that cater to users of high-speed Internet connection or Local Area Network (LAN). To install the application, the end-user only has to click the icon on the Web page or double-click an icon on the file share. Applications that are deployed using this setting are automatically added to the Start menu and the Add/Remove Programs menu.

**To install from a CD or DVD**—Deploys an application on a removable disk, such as CD or DVD. This setting is best suited for applications that would be used by users who have irregular or low-bandwidth network connections. Note that in this case, you do not need any network connection to execute the application but you do need a connection for updating the application.

#### 7. What are the enhancements in ClickOnce deployment in .NET 4.0?

- In .NET 4.0, the **ClickOnce** deployment technology is enhanced with the following features:
  - Support for .NET Framework 4.0 version**—Creates applications by using Visual Studio 2010 that can target .NET Framework 4.0 and its new features.
  - Support for multiple versions of the .NET Framework**—Creates applications that are compatible with multiple versions of the .NET Framework. You can specify the target framework for an application as .NET Framework 3.5 or .NET Framework 4 while creating the application.
  - Enhanced logging feature**—Stores logging information that includes various parameters passed to the ClickOnce runtime, the browser settings, and ClickOnce security options.
  - Custom Installer and User Interface**—Allows you to create a custom graphical user interface for installing and updating the .exe applications. In addition, the custom installer can have custom dialog boxes for security and maintenance operations.

#### 8. What is the difference between deploying and publishing an application?

- In deployment, you can create a new setup and deployment project. In this project, you can add the project output and create a setup.exe file. After creating an executable file, you need to login into the server and execute the setup.exe file to install the application. On the other hand, in publishing, you need to **right-click** the application in the **Solution Explorer** and **Select Publish** to publish the application. Then, you specify a location where the application is to be published. The users can then install the application from the location where you have published it and run locally even when the computer is offline.

## 9. How can you determine whether you should deploy the application or publish the application?

- If you want to host the application on a shared hosting environment, you should use publishing; whereas, if you want to create a Web application that is downloaded from a Web site, you should deploy the application to create a setup.exe file.

## 10. Explain the .NET Framework deployment features.

- In a general context, .NET Framework includes the following deployment features:
  - **No-impact applications** —Provides application isolation and removes DLL conflicts
  - **Private components by default** —Enables the components to deploy to the application directory and to be visible only to the containing application
  - **Side-by-side versioning** —Enables you to select one of the multiple versions
  - **XCOPY deployment and replication** —Refers to the self-descriptive application that is deployed without the need to store registry entries
  - **On-the-fly updates** —Allows for the updating of the DLLs of the remote computers
  - **Integration with the Microsoft Windows Installer** —Makes the features, such as advertising, publishing, repairing, and install-on-demand available during deployment of an application
  - **Enterprise deployment**—Eases the task of software distribution
  - **Downloading and caching** —Specifies that the downloads are kept smaller and the components are isolated for application use
  - **Partially trusted code**—Enables code-based identification

## 11. What do you mean by Merge Module projects?

- **Merge Module** projects are used to package the files and components that are shared between multiple applications. The Merge Module project file contains the **.msm** extension. The **.msm** file includes files, resources, registry entries, and setup logic. This file is merged with a Windows installer (.msi) file to correctly install the shared files. If a single merge module is used by more than one application, then you need to add that merge module in the package only once.

## 12. What is the need of Copy Web Site?

- Copy Web Site is a tool used to deploy the Web site by copying its content files. The Copy Web Site tool also checks whether or not the latest version of a file is present at the destination. If files of the most recent version are found at the destination, then the Copy Web Site tool does not superimpose the older version of files. The Copy Web Site deployment tool consists of the following main entities:
  - **Project source**—Specifies the source directory, which contains the contents and references of a Web site at development time. In simple words, you can say that the project source specifies the site that you currently have opened in Visual Studio 2010. The Copy Web Site tool picks all the files for deployment from this location.
  - **Project destination**—Specifies the destination folder where you have to deploy the application. This destination directory can be placed on remote computers or servers, which allow you to copy the Web site contents using the Front Page Server Extensions, FTP, or HTTP protocol implementations for content transfer.
  - **Synchronizing two Web sites**—Synchronizes two Web sites by copying each other's files. Synchronization checks the files on the local and remote sites and ensures that all files on both sites are up to date.
- You can deploy an ASP.NET Web application using either the Windows Installer deployment or ClickOnce deployment technique.

## 13. How can you deploy an ASP.NET Web application?

- You can deploy an ASP.NET Web application using either the Windows Installer deployment or ClickOnce deployment technique.

## 14. Can you deploy an ASP.NET Web application project using the Copy Web Site option?

- No. The Copy Web Site option can only be used to deploy the Web sites.

## 15. Can Windows applications and the Web applications be deployed using the same template of Setup and Deployment project?

- No. the Windows applications use the Setup Project template; whereas, the Web applications use the Web Setup Project template. After the deployment, their installation takes place in the similar way.

## 16. What is the use of the Copy Project command?

- The **Copy Project** command copies only the files required to run the project and pastes it on the target server. It does not deploy the

complete project; therefore, IIS directory settings are not automatically configured.

# 16

## 17. What is Application Cache?

- When a ClickOnce application is installed locally or hosted online, it is stored in the ClickOnce application cache of the client computer. The ClickOnce application cache is a set of hidden directories placed under the Local Settings directory of the current user's Documents and Settings folder. The application cache contains all the application files, assemblies, configuration files, application and user settings, and data directory. In case the ClickOnce applications are hosted online, the size of the ClickOnce application cache gets limited to a specified amount; whereas, the installed applications do not restrict to the cache size limitation. The cache storage quota is responsible to determine the size of the application cache.

## .NET Assemblies

### 1. What is an assembly?

- Assemblies are the basic building blocks required for any application to function in the .NET realm. They are partially compiled code libraries that form the fundamental unit of deployment, versioning, activation scoping, reuse, and security. Typically, assemblies provide a collection of types and resources that work together to form a logical unit of functionality. They are the smallest deployable units of code in .NET. Compared to the executable files, assemblies are far more reliable, more secure, and easy to manage. An assembly contains a lot more than the Microsoft Intermediate Language (MSIL) code that is compiled and run by the Common Language Runtime (CLR). In other words, you can say that an assembly is a set of one or more modules and classes compiled in MSIL, and metadata that describes the assembly itself, as well as the functionalities of the assembly classes.

### 2. Name the different components of an assembly.

- An assembly is a logical unit that is made up of the following four different types of components:
  - Assembly manifest
  - MSIL source code
  - Type metadata
  - Resources

### 3. What are the different types of assemblies? Explain them in detail.

- The following are the two types of assemblies:
  - Private Assembly* —Refers to the assembly that is used by a single application. Private assemblies are kept in a local folder in which the client application has been installed.

- Public or Shared Assembly** —Refers to the assembly that is allowed to be shared by multiple applications. A shared assembly must reside in Global Assembly Cache (GAC) with a strong name assigned to it.

For example, imagine that you have created a DLL containing information about your business logic. This DLL can be used by your client application. In order to run the client application, the DLL must be included in the same folder in which the client application has been installed. This makes the assembly private to your application. Now suppose that the DLL needs to be reused in different applications. Therefore, instead of copying the DLL in every client application folder, it can be placed in the global assembly cache using the GAC tool. These assemblies are called shared assemblies.

#### 4. Can one DLL file contain the compiled code of more than one .NET language?

- No, a DLL file can contain the compiled code of only one programming language.

#### 5. What is the maximum number of classes that can be contained in a DLL file?

- There is no limit to the maximum number of classes that can be contained in a DLL file.

#### 6. What is Assembly Manifest?

- Assemblies maintain all their information in a special unit called the manifest. Every assembly has a manifest. Table 16.1 shows the information contained in an assembly manifest.

Table 16.1: Contents of an Assembly Manifest

Assembly Manifest Content	Description
Strong name information	Represents the public key from the publisher, if a strong name is assigned to an assembly.
Type reference information	Represents a hash of each file contained in the assembly and its file name. Represents the information used at the runtime to map a type reference to the file that contains its declaration and implementation.

#### 7. What is metadata?

- The first four items—the assembly name, version number, culture, and strong name information—make up the assembly's identity.

- An assembly metadata describes every data type and member defined in the code. It stores the description of an assembly, such as name, version, culture, public key of an assembly along with the types exported, other assemblies dependent on this assembly, and security permissions needed to run the application. In addition, it stores the description of types, such as the name, visibility, base class, interfaces implemented, and members, such as methods, fields, properties, events, and nested types. It also stores attributes. Metadata is stored in binary format. Therefore, metadata of an assembly is sharable among applications that execute on various platforms. It can also be exported to other applications to give information about the services and various features of an application.
- The AssemblyInfo.cs file stores the assembly configuration information and other information, such as the assembly name, version, company name, and trademark information. (True/False).
  - True.

Version number	Represents a major and minor version number, as well as a revision and build number. The CLR makes use of these numbers to enforce version policy.
Culture	Represents information of the culture or language, which the assembly supports. An assembly is a container of only resources containing culture- or language-specific information.

- 9. What is GAC?**
- GAC is a central repository (cache) in a system in which assemblies are registered to share among various applications that execute on local or remote machines. .NET Framework provides the GAC tool (gacutil.exe utility), which is used to view and change the content of GAC of a system. Adding new assemblies to GAC and removing assemblies from GAC are some of the tasks that can be performed by using the gacutil.exe utility. GAC can contain multiple versions of the same .NET assembly. CLR checks GAC for a requested assembly before using information of configuration files.

The gacutil.exe /i <assembly name> is the command that is used to install an assembly in GAC. Users use the Command Prompt of Visual Studio to install an assembly in GAC by using this command.

You can see all the assemblies installed in the GAC using the GAC viewer, which is located at the <WindDir>\assembly directory, where <WindDir> is WINDOWS in Windows XP or Windows in Windows Vista or WINNT in Windows 2000. Apart from the list of assemblies, the assembly viewer also shows relevant information, such as the global assembly name, version, culture, and the public key token.

- 10. What is the value of the Copy Local property when you add an assembly in the GAC?**
- False.

- 11. Is versioning applicable to private assemblies?**

- No, versioning is not applicable to private assemblies as these assemblies reside in their individual folders. Versioning can be applied to GAC only.

- 12. Discuss the concept of strong names.**

- Whenever, an assembly is deployed in GAC to make it shared, a strong name needs to be assigned to it for its unique identification. A strong name contains an assembly's complete identity—the assembly name, version number, and culture information of an assembly. A public key and a digital signature, generated over the assembly, are also contained in a strong name. A strong name makes an assembly identical in GAC.

- 13. What is the significance of the Strong Name tool?**

- The Strong Name utility (sn.exe) helps in creating unique public-private key pair files that are called strong name files and signing

- assemblies with them. It also allows key management, signature generation, and signature verification.

- 14. Where is the information regarding the version of the assembly stored?**

- Information for the version of assembly is stored inside the assembly manifest.

- 15. Name the MSIL Disassembler utility that parses any .NET Framework assembly and shows the information in human readable format.**

- The Ildasm.exe. utility.

- 16. What is Native Image Generator?**

- The Native Image Generator (Ngen.exe) is a tool that creates a native image from an assembly and stores that image to native image cache on the computer. Whenever, an assembly is run, this native image is automatically used to compile the original assembly. In this way, this tool improves the performance of the managed application by loading and executing an assembly faster.

Note that native images are files that consist of compiled processor-specific machine code. The Ngen.exe tool installs these files on to the local computer.

- 17. Which utility allows you to reference an assembly in an application?**

- An assembly can be referenced by using the gacutil utility with the /r option. The /r option requires a reference type, a reference ID, and a description.

- 18. How do you add and remove an assembly from GAC?**

- You can add and remove an assembly from GAC by using the Gacutil utility.

You can add an assembly to GAC by using the following syntax:

```
gacutil /u [AssemblyName] /assembly[Path]
```

You can remove an assembly from GAC by using the following syntax:

```
gacutil /u [AssemblyName] /assembly[Path]
```

- 19. What is a satellite assembly?**

- Satellite assemblies are assemblies that are used to deploy language and culture specific resources for an application. In an application, a

separate product ID is assigned to each language and a satellite assembly is installed in a language specific sub-directory.

#### 20. How can different versions of private assemblies be used in the same application without a re-build?

- You can use different versions of private assemblies in the same application without a re-build by specifying the assembly version in the AssemblyInfo.cs or AssemblyInfo.vb file.

#### 21. What is the difference between EXE and DLL?

- Table 16.2 shows the difference between EXE and DLL:

EXE	DLL
It is an executable file, which can be run independently.	It is Dynamic Link Library that is used as a part of EXE or other DLLs. It cannot be run independently.
EXE is an out-process component, which means that it runs in the application process (process space).	It runs in the application process (memory), so it is called as in-process component.
It cannot be reused in an application.	It can be reused in an application.
It has main function.	It does not have a main function.

## Remoting and Reflection

#### 1. Explain .NET remoting.

■ .NET remoting provides a powerful programming model and runtime support to make objects available outside the boundaries of machine as well as the processing boundaries of an application. It allows a developer to build widely-distributed applications, irrespective of the fact whether the components of the application are on the same computer or spread over the network. .NET remoting uses the concept of client and server applications.

Data can be transmitted over any of the protocol like Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Transmission Control Protocol (TCP).

#### 2. Define activation.

- Activation is a process of initializing a remotable object before its instantiation.

#### 3. What is a Windows Process?

- A Windows process is an instance of an application running in the background when the Windows starts.

#### 4. How .NET remoting is different from Distributed Component Object Model (DCOM)?

■ DCOM is suitable for applications that exist on the same computer and are of the similar type; whereas, .NET remoting allows you to build widely distributed applications. Another difference is that DCOM relies on the binary protocol only, which is not supported by all the object models. On the other hand, .NET remoting uses binary, TCP, HTTP, and Simple Object Access Protocol (SOAP) protocols to send and receive the data over a network.

#### 5. What do you mean by an application domain?

- An application domain is a virtual location in memory where a process runs. Prior to .NET, all applications were executed in Win32 processes

and a separate entry had to be created for every running application. Each process has its own private virtual memory and writing one process to another process's memory would lead to system crash. AppDomains are separated from one another in much the same way the Win32 processes were. However, unlike Win32 processes, .NET has CLR to see that AppDomains are performing according to their defined boundaries. CLR makes sure that application code is type safe and is not pointing outside the boundaries of the application at the assembly load up time itself.

Figure 17.1 shows that AppDomains stay within the .NET remoting system:

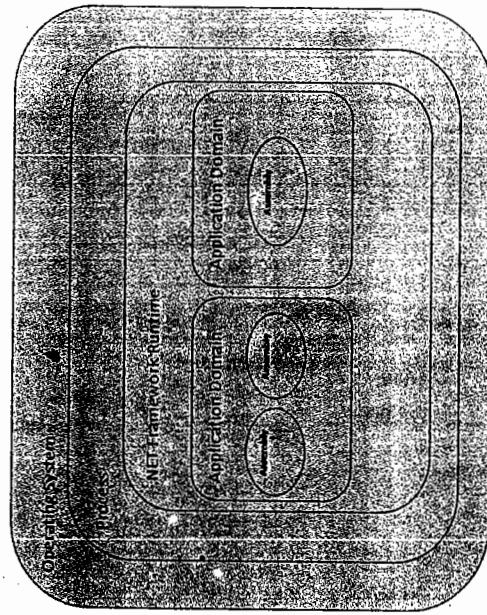


Figure 17.1: Application Domains

6. List the activation modes in which you can create a remote object.

- The following are the two different activation modes in which you can create a remote object:
  - Server Activation
  - Client Activation

7. Explain the .NET remoting architecture.

- .NET remoting provides a flexible architecture for distributed applications. It enables objects within different application domains to communicate together. However, .NET remoting also enables communication between objects when their application domains are

separated across the network. Figure 17.2 displays the architecture of .NET remoting:

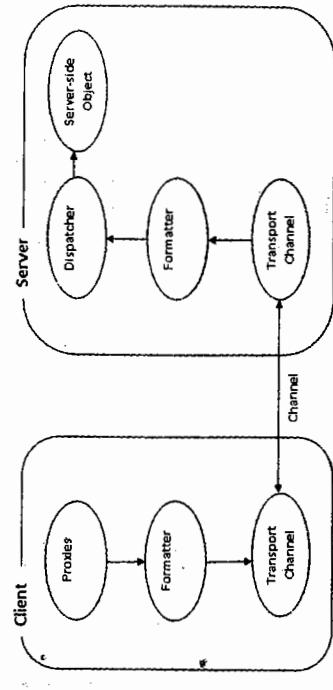


Figure 17.2: .NET Remoting Architecture

.NET remoting uses the concept of proxy, which is an abstraction of the real object that can be treated, as if it were a real object. When a client object wants to create an instance of the server object, the remoting system at the client side creates a proxy of the server object at the client-side. However, the behavior of the proxy object does not change as it behaves like a remote object.

The remoting system on the client sends a call over the channel to the remoting system on the server. Before sending the call to the server remoting system, the object is serialized into a specific transfer format by using a formatter, such as SOAP or binary. The serialized message later reaches a transport channel that transfers it to the remote system by using a specific protocol, such as HTTP or TCP. The remoting system on the server then handles the request and passes it to the formatter that deserializes the message back into its original format and forwards it to the dispatcher. The dispatcher calls the target object's method and forwards the message to the server object for taking the appropriate action. The result is then passed to the server remoting system from the server object, which in turn, passes it to the remoting system on the client through the communication channel.

8. What is the difference between remotable and nonremotable objects?

- Remotable objects are those that can be accessed outside their own AppDomain using a proxy. On the other hand, nonremotable objects are those that cannot be accessed outside their own AppDomain, which means instances of a nonremotable object cannot cross .NET remoting boundaries. However, if you attempt to pass an instance of a nonremotable object to another AppDomain, an exception will

occur. The nonremotable objects are designed for use within the same AppDomain in which they were created and are always accessed directly from that AppDomain. In addition, the nonremotable objects cannot be copied or represented in another AppDomain.

#### 9. Define marshalling.

- Marshalling is a process of packaging and sending method calls between the objects and across application boundaries by using serialization and deserialization.

#### 10. Differentiate between the marshal-by-value and marshal-by-reference objects.

- The marshal-by-value (MBV) objects declare their serialization rules by either implementing the ISerializable interface or by being marked with the SerializableAttribute attribute, which in turn, serializes the object automatically. The marshal-by-value objects are copied and passed from the server AppDomain to the client AppDomain. The ISerializable interface belongs to the System.Runtime.Serialization namespace. It provides the GetObjectData() method to populate the SerializationInfo object with the data that has to be serialized in a remote object.

The marshal-by-reference (MBR) objects are accessed on the client side using a proxy. The client only needs to hold the reference of the objects. When a client creates an instance of the MBR object in its own application domain, the .NET Remoting Framework creates a proxy to represent the MBR object and sends a reference of that proxy to the client. The client then makes calls on the proxy. .NET remoting marshals those calls to the application domain of the remote object and invokes the call. To define an MBR object, you need to use the System.MarshalByRefObject class.

#### 11. Do marshal-by-value objects require an activation process?

- Marshal-by-value objects do not require any activation process because they are copied using the serialization process and activated using deserialization.

#### 12. What are well-known objects?

- Server-activated objects that are published at a URL outside the application domain are called well-known objects.

#### 17. Describe client-activated objects (CAOs).

- The client-activated objects (CAOs) are objects whose lifetime is directly controlled by the client. To handle well-known CAOs, .NET Remoting Framework assigns a Uniform Resource Identifier (URI) to each instance of a CAO object. The CAO objects are created on the server corresponding to the client requests. A CAO instance serves

#### 13. What are context-bound objects?

- The context-bound objects are further refinement of marshal-by-reference objects. These objects restrict instances of their objects to remain within a specific context. Objects external to the containing context cannot directly access these objects, even if the other objects are within the same application domain.

#### 14. What do you mean by server-activated objects?

- Server-activated objects (SAOs) are remote objects whose lifetime is controlled by the server. You cannot create objects by using the new keyword. An SAO object is activated when a client calls a method on the proxy object. .NET Remoting Framework refers to these objects as well-known objects, because the server application publishes these objects at a well-known Uniform Resource Locator (URL) before activating object instances.

#### 15. What are the two modes to activate the SAOs?

- The following are the two modes to activate server-activated objects:
- Singleton mode
- SingleCall mode

#### 16. What is the difference between the Singleton mode and SingleCall mode?

- Using the Singleton mode, one object can serve the requests of all clients. An instance of the Singleton object is created when the first client tries to access it. After that, the subsequent client calls are channeled to the same Singleton object. The Singleton object maintains the state across requests or method calls. This state is globally shared between all the clients.

The SingleCall mode instantiates an object to serve a single client request. When you configure an object as SingleCall, the .NET Remoting Framework activates a new instance of that type for every method invocation a client makes, after which it is garbage collected for recycling. Since the SingleCall mode handles only one request, it does not store state between requests and is called stateless. The SingleCall mode potentially allows for greater server scalability.

only the client that creates it and does not get discarded with each request. It also enables the CAO to maintain state for each client that it is serving; however, it cannot share the common state. The CAOs are mostly used when the client needs to maintain a private session with the remote object.

#### 18. What is a channel? List different types of channel.

- Channels are the devices used to build communication across remote boundaries. .NET Remoting architecture ensures that at least one channel should be registered before registering an object on the remote system. In addition, a client object must specify a channel to communicate with the remote object. The following are the three types of channels:

- TCP channel
- HTTP channel
- IPC channel

#### 19. Explain the TCP channel.

- The TCP channel uses the TcpChannel class to set up a remoting server and its client. It is a socket-based transport that uses the TCP protocol for transporting the serialized message stream across .NET remoting boundaries. The TcpChannel class is defined in the System.Runtime.Remoting.Channels.Tcp namespace and implements the IChannel, IChannelSender and IChannelReceiver interfaces, which means that the TcpChannel class supports both sending and receiving data across .NET remoting boundaries. By default, the TcpChannel class uses binary formatter but it is not bound to use only this one. Instead, the TcpChannel class can use any of the two formatters, binary or SOAP.

#### 20. What is the difference between HTTP and IPC channels?

- The HTTP channel uses the HttpChannel class to implement a client channel for remote calls. This class uses the HTTP protocol to provide maximum interoperability and to transfer messages across the Internet and through firewalls. The HttpChannel class can be used over the Internet because firewalls generally do not block HTTP communication. The HttpChannel class is defined in the System.Runtime.Remoting.Channels.Http namespace and implements the HTTP transport functionality. Similar to the TcpChannel class, the HttpChannel class can also send and receive messages across .NET remoting boundaries. By default, the HttpChannel class uses the SOAP formatter to serialize messages; however, it can also use the binary formatter. The HttpChannel class

is more effective than the TcpChannel class from security point of view, as it offers immediate accessibility to security features of HTTP server, such as the IIS (Internet Information Services) offers SSL (Secure Sockets Layer) and Windows authentication.

The Inter Process Communication (IPC) channel uses the IpcChannel class to implement a channel that uses the IPC protocol to transmit messages. The IpcChannel class is much faster than the TcpChannel and TcpChannel1 classes, as it bypasses the traditional network communication to cross AppDomains. Therefore, the IpcChannel1 class is used only for communication between application domains on the same computer. Moreover, you cannot use the IpcChannel class to build a distributed application that spans multiple physical computers. The IpcChannel class is defined in the System.Runtime.Remoting.Channels.Ipc namespace and implements the functionalities of the IpcClientChannel and IpcServerChannel classes. However, the IpcChannel class can also send and receive messages across .NET remote boundaries.

#### 21. What are message sinks?

- Message sinks are the objects that are used to transfer a message from a client application to a server-side object. A sink usually receives a message from another object, processes that message, and passes it on to the next sink in the chain. Sink provides security by encrypting the data or messages before sending it. All message sinks implement the IMessageSink interface, which has a single property, NextSink, and two methods, SyncProcessMessage() and AsyncProcessMessage().

#### 22. What are different types of message sinks?

- Message sinks can be categorized as follows:
  - Envoy Sink*—Enables a server to pass messages to a client. It is mainly created from the server context and used to collect client information and pass it to the server.
  - Server Context Sink*—Receives the messages from server-side.
  - Server Object Sink*—It is created for a particular object if its class defines *context sinks and attributes*.
  - Client Context Sink*—It is created by any object that is within the context on objects located outside the context.

### 23. What are formatters? List different types of formatters.

- Formatters are objects that encode and serialize data into the appropriate format before transmitting it over a channel. The following are the two types of formatters:
  - Binary formatter
  - SOAP formatter

These are implemented by using the `BinaryFormatter` and `SOAPFormatter` classes.

### 24. Give a brief description about `BinaryFormatter` and `SOAPFormatter`.

- The `BinaryFormatter` class converts an object's state into a binary stream. This class is used for network transportation that allows you to send and receive binary data, such as TCP/IP. The `BinaryFormatter` class is defined in the `System.Runtime.Serialization.Formatters.Binary` namespace. The `BinaryFormatter` class is used by an object of the `BinaryClientFormatterSinkProvider` class to serialize messages for the client channel through which remote messages flow. An object of the `BinaryClientFormatterSinkProvider` class represents a client formatter sink that provides necessary properties to serialize the message to the stream.

The `SOAPFormatter` class converts an object's state into a SOAP format. SOAP is an XML-based protocol that is used to exchange information between applications. As some network transports do not allow you to send and receive binary data; therefore, they force applications to convert all binary data into an American Standard Code for Information Interchange (ASCII) text representation before sending it over a channel. In such situations, the `SOAPFormatter` class is used to send and receive data. The `SOAPFormatter` class is defined in the `System.Runtime.Serialization` namespace. The `SOAPFormatter` class is used by an object of the `SoapClientFormatterSinkProvider` class to serialize messages for the client channel through which remote messages flow. An object of the `SoapClientFormatterSinkProvider` class represents a client formatter sink that provides necessary properties to serialize the message to the stream:

### 25. What do you mean by proxies?

- The proxy acts as a local representative of the remote object residing in an external AppDomain. They receive all the calls on the behalf of the remote objects and route them to the correct remote object.

### 26. Define transparent and real proxies.

- The **transparent proxy** creates the illusion that the actual object resides on a client when the client and server are in the same application domain. The transparent proxy interacts with a client and captures the remote method call made by the client. The transparent proxy is implemented by using the `TransparentProxy` class. The real proxy creates the illusion that the actual object resides on a client when the client and server are in different application domains. The real proxy actually sends and receives the messages from a remote object. A real proxy is implemented by using the `RealProxy` class.

### 27. How can you distinguish between Web services and remoting?

- Web services and .NET remoting can be distinguished based on the different capability factors given in Table 17.1.

Table 17.1: Distinguishing Factors between Web Services and Remoting

Capability	Web Services	Remoting
Invoking single method on a stateless object	Yes	Yes
Invoking multiple methods on a stateful object	No	Yes
All clients invoking methods on the same server-side object	No	Yes
Passing through firewalls	No	No
Using HTTP for communication	Yes/No	Yes
Using raw or standard communication	No	Yes
Using IIS as host	Yes	Yes
Allowing custom hosts	No	Yes
Using SOAP-compliant formatting of data	Yes	Yes

Capability	Web Services	Remoting
Retrieving partial copy of data from complex object	Yes	Yes
Retrieving complete copy of complex objects	No	Yes

**28. How can you decide to use the singleton mode in remoting?**

- If all the remoting clients share the same data, you can use the Singleton mode.

**29. Are client-activated objects stateful in nature?**

- Yes. In the CAO remoting model, a client creates an instance on the server. The instance serves only the client that creates it and does not get discarded with each request. This enables the CAO to maintain state for each client it is serving; however, the instance cannot share a common state.

**30. Which configuration file contains all the supported channels?**

- The Machine.config file contains all the supported channels.

**31. Is it appreciated to distribute the implementation to remoting clients? If no, why?**

- No. You should not distribute the implementation to client due to the following reasons:
  - Security — Anyone can use the manifest (ILDASM) and decrypt the logic
  - Redistribution — If you make any changes in application, you need to redistribute the complete implementation again

**32. How is a method called asynchronously in remoting?**

- Using delegates, you can call a method asynchronously in remoting.
- The MarshalByRefObject class is inherited while making an object remotable.

**33. Which class do we inherit while making an object remotable?**

- The RegisterWellKnownServiceType() method helps in defining and configuring a remotable object that needs to be activated on the server side. This method takes the following three parameters:

**34. What are the different ways to host a remote object?**

- A remote object can be hosted in any one of the following hosts:
  - Managed & executable or Windows service
  - IIS
  - .NET component services

**35. What do you mean by the lifetime (lease time) of a remote object?**

- Once the object is published and has started interacting with external AppDomains, a decision has to be made about the lifetime (lease time) of the remote object so that the object could be revoked at some point of time when it has served enough or is no longer being used by anyone. The concept of lease time applies to a singleton or client activated object only. By default, the time of five minutes is specified to a client object. However, you can change this default time in the constructor of your remote object.

**36. How is the lease of an object defined?**

- To define the lease<sup>®</sup> of an object, implement the `ILease` interface while writing the class code.

**37. Is it possible to configure a .NET Remoting object by using an XML file?**

- Yes, it is possible. You can do so with the help of the `machine.config` and `web.config` in ASP.NET.
- The `Soapsuds` tool is used to automatically generate interface for the remotable object in .NET.

**39. What is WSDL?**

- WSDL stands for Web Service Description Language. It helps display the Web service information to the consumer.

**40. What is the difference between the `RegisterWellKnownServiceType()` and `RegisterWellKnownClientType()` methods in .NET remoting?**

- The `RegisterWellKnownServiceType()` method helps in defining and configuring a remotable object that needs to be activated on the server side. This method takes the following three parameters:

```
RegisterWellKnownClientType(<Class>/objectName>,<:  
wellKnownObjectMode.Singleton/singleCall>);
```

On the other hand, the RegisterWellKnownClientType() method is used on the client side and helps in communicating with the server-side activated remote object. This method takes two parameters as follows:

```
RegisterWellKnownClientType(<Class>/objectName  
<>),<"Server Remotable Object URL">;
```

#### 41. Do any security measures exist for .NET Remoting in the System.Runtime.Remoting namespace?

- No, security should be taken care of at the application level. Cryptography and other security techniques are applicable at the application or server level.

#### 42. What is reflection?

- Reflection is the process of finding out the internals, such as metadata, assemblies, modules, and types of an application without accessing the source code. You can also use reflection to find all classes in an assembly and all the methods, properties, and events that might be supported by each of those classes. Reflection also allows you to search for all kinds of information about a class, such as the base class from which it is derived and the interfaces it supports.

#### 43. Why do we need reflection?

- Reflection is an important feature of .NET Framework and is very useful when large applications are developed. It is also necessary because it has the ability to obtain information at runtime about what is loaded and how the members of the classes that are loaded will be called. In other words, reflection is necessary for adequate versioning, for dynamic loading of classes, and for proper security handling.

#### 44. How are application, process, and application domains are related?

- An application is an executable file on the hard drive or network. A process is an instance of a running application. Numerous processes can be launched by the same application; however, one process can run just one application. Application domain is a virtual location in memory where a process runs.

#### 45. When will you use .NET Remoting and when will you use ASP.NET Web Services?

- Remoting is used when want to control both ends of an application that is involved in the communication process. Web Services are used when you are either a client or a server and another end is being controlled by someone else.

#### 46. Which classes are used by reflection to get information about various types?

- Reflection uses the System.Reflection namespace that contains the following important classes:
  - Assembly —Retrieves and manipulates information about an assembly. This class allows you to perform various functions, such as defining and loading assemblies, loading modules that are listed in the assembly manifest, and locating a type from this assembly to create an instance of it. The Assembly class uses one of the two methods—Assembly.GetType() or Assembly.GetTypes()—to get the Type object from assemblies that has not been loaded.
  - Module class —Retrieves information about parent assembly, classes in the module, and all the global methods available in the classes.
  - ConstructorInfo —Retrieves the information about a constructor, such as constructor name, parameters, access modifiers, and various implementation details. The ConstructorInfo class uses the GetConstructor or GetConstructors method of a Type object to retrieve the constructor information.
  - MethodInfo —Retrieves information about a method in the class, module, or assembly, such as method name, return type, parameters, access modifiers, and implementation details. The MethodInfo class uses either the GetMethod or GetMethods method of a type to retrieve the information.
  - FieldInfo —Retrieves information about the fields, such as name, access modifiers, and implementation details. In addition, you can use this class to get or set field values of a class.
  - EventInfo —Retrieves information about events, such as name, event-handler data type, custom attributes, declaring type, and reflected type.

- ParameterInfo** —Retrieves information about a parameter, such as its name, data type, whether it is an input or an output parameter, and its position in a method signature. This class uses the `GetParameters` method to return an array of `ParameterInfo` objects that represents the parameters of a method.

# 18

## ASP.NET MVC Framework

### 1. What is Model-View-Controller (MVC)?

- While developing an Enterprise application, it is logical to interweave data access, business logic, presentation logic, and control together. However, for a large enterprise-level application, interweaving makes code files more complex; therefore, it will be difficult to manage while implementing changes in future. This results in the duplication of data access and business logic code in the application at various points. As a result, the implementation, testing, and maintenance efforts are also duplicated. It is always a good idea to reduce duplication by separating presentation logic, business logic, and control logic as different components, which can easily be reused. The solution to these problems comes in the form of a Model-View-Controller (MVC) design pattern, which separates data access, business logic, and presentation logic and enables creation of a number of views accessing the same enterprise application model.

### 2. What is ASP.NET MVC Framework?

- The ASP.NET MVC Framework is not a substitute for the ASP.NET Web Forms pattern. It just provides an alternative choice to the developer while designing and developing a Web application. It enables you to achieve and maintain a clear separation of presentation layer (UI), logic, and data access. It also facilitates test-driven development (TDD) environment, where you can implement automated unit tests, which define and verify the requirements of new code before you actually write the code itself.
- It is a lightweight and extremely testable presentation framework that is appended to existing ASP.NET features, such as master pages and membership-based authentication. The MVC framework resides inside the `System.Web.Mvc` namespace. This namespace is an elementary supported part of the System.Web namespace, which forms the basis of any Web application.

### 3. What are the features of ASP.NET MVC Framework?

- The MVC-based ASP.NET architecture has various advantages over the traditional approach of developing Web applications. ASP.NET MVC Framework is loaded with the following features:

- MVC Framework separates the application tasks, such as input logic, business logic, and UI logic, testability, and test-driven development (TDD), by default. TDD is a method of software development in which unit testing is done repeatedly on the source code.
- ASP.NET MVC Framework's components are designed in such a way that they can be easily replaced or customized. It enables you to make use of Dependency Injection (DI) and Inversion of Control (IoC) container models. DI enables you to inject objects into a class. IoC enables you to use an object from outside, such as configuration file, if another object requires it.
- ASP.NET MVC Framework supports markup files as view templates in the existing ASP.NET page (.aspx files), user control (.ascx files), and master page (.master files). You can make use of ASP.NET features with ASP.NET MVC Framework also as an add-on, such as nested master pages, declarative server controls, templates, data binding, and localization.
- ASP.NET MVC Framework supports the existing features of ASP.NET, such as forms authentication and Windows authentication, URL authorization, membership and roles, output and data caching, session and profile state management, health monitoring, configuration system, and the provider architecture.
- ASP.NET MVC Framework provides complete control over the HTML markup code.
- ASP.NET MVC Framework helps you in creating applications by integrating AJAX and jQuery.

#### 4. Explain the components of ASP.NET MVC Framework.

- The following are the components of ASP.NET MVC Framework:
- *Model in MVC Framework* —The Model in MVC Framework represents the data and the business logic of the application, and is not related to the presentation of the application. The model components focus on keeping track of the state of your application. A Model can be considered as the foundation of the MVC architecture. Its core is composed of business logic and data access code. For example, if an application has to find out the gross salary of employees, the computation part is performed in the Model layer.
- The View and Controller interact with the Model for performing any task. In the MVC architecture, the Model layer is self-contained and functions independent of the View and Controller layers. Therefore, the core application code, which is based on the Model, can be used with multiple user interfaces.

#### 5. Explain the interaction between the three components of the MVC architecture.

- The interaction between the three components of MVC architecture—Model, View, and Controller, is as shown in Figure 18.1:

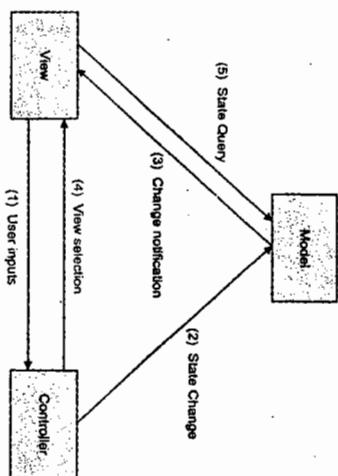


Figure 18.1: Interaction between the MVC Architecture Components

Each user may get a different view displaying the required information that is customized for the value and layout according to that particular user. The application updates the View as soon as any change in the data takes place. If the user wishes to change the data, it sends a request to the Controller, which further consults the Model to update the View. It is to be noted that the View does not communicate directly with the Model; rather this communication

- *View in MVC Framework* —The View provides the UI for the Model. The user interacts with the application through a View. It represents the information based on the Model and allows the user to alter the data. Typically, this UI is created from the model data. The main responsibilities of a View are as follows:
- Rendering the content of the Model
  - Requesting the updates from the Model
  - Sending user inputs to the Controller
  - Allowing the Controller to select the View
- *Controller in MVC Framework* —Controllers are the components that handle all the Views associated to a Model, and ultimately select a view to render that displays an UI. When the user interacts with the View and tries to update the Model, the Controller invokes methods to update the Model. The Controller controls the data flow and transformation between the Model and the View. Its main responsibilities are as follows:
- Mapping the user actions into Model updates
  - Selecting the View for responses

takes place through the Controller. The control flow can be broken down into the following steps:

- The user sends a request through a UI provided by the View, which further passes this request to the Controller.
- The Controller receives the input request coming from the user through the interface provided by the View and processes the request according to the Controller logic. If access to the Model is not required, Controller moves to step 4; otherwise, the Model is accessed, and its state is changed.
- If the Model is accessed and modified, all the associated Views are notified of the changes.
- The Controller then selects the new View to be displayed.
- The View presents a UI according to the Model. The View data is also contained within the Model. View queries about the state of the Model to show the current data.
- The View remains idle after the current interaction and waits for the next interaction to begin. The entire algorithm is repeated when a new request is serviced in the same way as for the other requests.

6. Which class defines the preconfigured URL routes for the ASP.NET MVC Framework?
- The Controller class is the base class of the ASP.NET MVC Framework.
  - The ASP.NET Application class defines the preconfigured URL routes for ASP.NET MVC Framework inside the Global.asax file.
7. Which class defines the preconfigured URL routes for the ASP.NET MVC Framework?
- The ASP.NET Application class defines the preconfigured URL routes for ASP.NET MVC Framework inside the Global.asax file.
  - The Custom Controller factories require new signatures due to changes in the DefaultControllerFactory class.

8. What is URL Routing in ASP.NET MVC?
- Routing validates variables used in the URL definition according to a pattern mentioned in it and automatically passes them as parameter arguments to a Controller action after validation.
  - In the MVC architecture, data (Model) and user interface (View) are separated from each other so that any change in the user interfaces does not affect the data in the database. (True/False)
  - True.

10. What is TDD environment?
- ASP.NET MVC Framework facilitates the Test-Driven Development (TDD) environment, where you can perform an automated unit test, which specifies and verifies the requirements of new code before you write the code itself.

11. ASP.NET MVC Framework provides an alternative to the developer while designing and developing a Web as well as Windows applications. (True/False)

- False.

12. Which is the latest version of ASP.NET MVC that is built-in in Visual Studio 2010?

- The latest version of ASP.NET MVC is ASP.NET MVC 2.

13. What are the changes introduced in the ASP.NET MVC 2 framework?

- The changes introduced in the ASP.NET MVC 2 framework are:
  - Change in property validation behavior for classes that implement the IDataErrorInfo interface.
  - IIS script-mapping script is no longer available in the MVC installer.
  - The Html.Substitute helper method in MVC is removed.
  - The IValueProvider interface replaces all the uses of IDictionary<string, ValueProviderResult>.
  - The new CSS classes are added in the Site.css file.
  - The HTML helpers can now return an MvcHtmlString object.
  - The JsonResult class in MVC responds only to the HTTP POST requests.
  - The ModelMetadata property is added to the ModelBindingContext class in MVC 2.
  - The Custom Controller factories require new signatures due to changes in the DefaultControllerFactory class.
  - Area is now a reserved route-value key and cannot be used with route URL.

14. Explain the new features included in MVC 2 Framework.
- The following are the significant new features of the MVC 2 framework:
    - Templated Helpers* —Enables building of user interface that is based on the data model.
    - Areas* —Groups controllers and views in MVC into sections of a large Web application having isolation from other sections.
    - Asynchronous Controllers* —Processes requests asynchronously.
    - DefaultRouteAttribute* —Allows the user to initialize the argument parameter of an action method with a default value.
    - Binding Binary Data* —Supports binding of binary data with Model binders.

- *DataAnnotations Attribute*—Enables you to implement validation logic to a Model by using metadata attributes
  - *Model-Validator Providers*—Provides an abstraction that implements validation logic for the Model
  - *Client-Side Validation*—Provides client-side validation by library included in ASP.NET MVC 2 framework. It consists of JavaScript Object Notation (JSON) serialized data.
  - *New Code Snippets for Visual Studio 2010*—Provides information about predefined HTML code snippets in Visual Studio 2010
  - *Overriding the HTTP Method Verb*—Determines which action to perform for a resource in a Web site, which is developed using the Representational State Transfer (REST) architectural style
  - *HtmlValidationSummary Method*—Displays only Model-level errors instead of showing all the validation errors
  - *T4 Templates*—Generates code and markup according to the version of the .NET framework
- 15. Explain the T4 template included in the ASP.NET MVC 2 Framework.**
- ASP.NET MVC 2 includes a new property for Text Template Transformation Toolkit (T4) Templates that is used to specify the version of the .NET framework of the Web application. This property allows T4 templates to generate code and markup according to the version of the .NET framework. In Visual Studio 2010, the value of the property can be either 3.5 or 4.0.
- 16. What are new improvements in ASP.NET MVC 2 API?**
- In the ASP.NET MVC 2 API, the following new members are included:
    - The `Controller` class includes protected virtual `CreateActionInvoker` method. The `Controller` class invokes the method by using the `ActionInvoker` property and allows the delay instantiation of the invoker if no invoker is already set.
    - The `AuthorizeAttribute` class includes protected virtual `HandleUnauthorizedRequest` method. This helps filters that are derived from `AuthorizeAttribute` to get control of the behavior when the authorization fails.
    - The `Add (string key, object value)` method is included provides the dictionary initializer syntax for `ValueProviderDictionary`.
    - The `Sys.Mvc.AjaxContext` class includes the `get_object` method. It returns the JSON object.
    - The `UriParameter.optional` token included that is used to solve problems while binding data to a Model

## Cloud Computing

# 19

### 1. What is cloud computing?

- The cloud computing is the computing which is completely based on the Internet. It can also be defined as the next stage in the evolution of the Internet. The cloud computing uses the cloud (Internet) that provides the way to deliver the services whenever and wherever the user of the cloud needs. Companies use the cloud computing to fulfill the needs of their customers, partners, and providers. The cloud computing includes vendors, partners, and business leaders as the three major contributors. The vendors are the one who provide applications and their related technology, infrastructure, hardware, and integration. The partners are those who offer cloud services demand and provide support service to the customers. The business leaders are the ones who use or evaluate the cloud service provided by the partners. The cloud computing enables the companies to treat their resources as a pool and not as independent resources.

### 2. What is a cloud?

- A **cloud** is a combination of hardware, networks, storage, services, and interfaces that helps in delivering computing as a service. It has broadly three users which are end user, business management user, and cloud service provider. The **end user** is the one who uses the services provided by the cloud. The **business management** user in the cloud takes the responsibility of the data and the services provided by the cloud. The **cloud service provider** is the one who takes care or is responsible for the maintenance of the IT assets of the cloud. The cloud acts as a common center for its users to fulfill their computing needs.

### 3. What are the basic characteristics of cloud computing?

- The four basic characteristics of cloud computing are given as follows:
  - Elasticity and scalability
  - Self-service provisioning and automatic de-provisioning

- Standardized interfaces
- Billing self-service based usage model

#### 4 Explain the cloud architecture.

■ The cloud has its architecture based on the services, which it provides to the end user. Different service provider companies have built different architecture of cloud, based on the requirements and demands of the clients. The commonly used cloud architecture consists of large data centers having management fabric, which acts as a communication medium between the server, the hardware, and the operating systems. In addition, it also automates the deployment of virtual operating system images on server hardware. The cloud data center consists of many servers of large storage capacity for storing operating system images. The management fabric controls the deployment process by allocating and de-allocating the hardware and operating system images, as and when required. When a user deploys the service on the cloud, the management fabric deploys the operating system images and the user's service on the servers. After the service is deployed on the servers, it can be used easily when requested by the client. The service can have multiple instances that can be created if the demand and the requirements of the service are high. The usage of the service is purely based on the billing model of the cloud.

Figure 19.1 shows the fundamental cloud data center architecture:

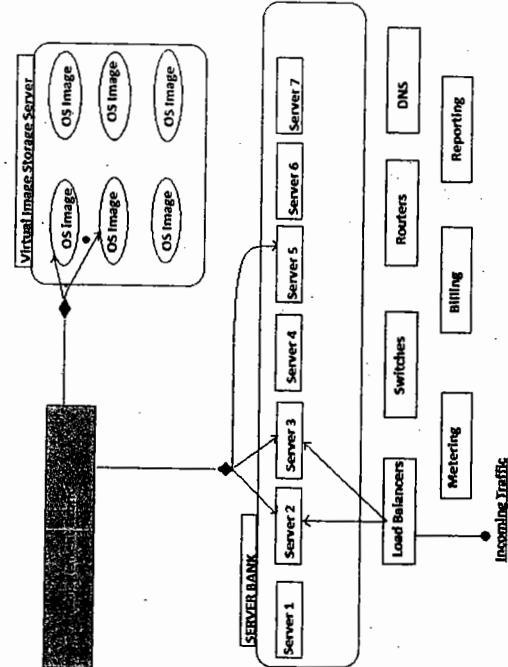


Figure 19.1: Displaying the Cloud Data Center Architecture

The cloud architecture includes some fixed hardware assets, such as load balancers, switches, routers, and Domain Name System (DNS) servers that help in the management of work load distribution across multiple instances of the server. The architecture also includes important components, such as metering, billing, and reporting that empowers the architecture with the ability to measure and report the usage of the service by a particular user. The measurement and reporting of the service are used for calculating the rent of the service by the client. The cloud can have geographically distributed data centers for providing geo-located services to provide data across the world. Due to geo-located services, the user can access the service from anywhere in the world to access the required data, whenever needed.

Different service providers may have different implementation and approach to this pattern. The cloud architecture is built by keeping in mind that it can provide generalized services to the client. The generalized service attracts a lot of customers; and therefore, raises the profits. The cloud architecture also ensures that the data of one customer must not be mixed with the other. The good cloud architecture helps to increase the number of customers faster than the traditional approach of developing the software for providing the services. The cloud saves tremendous amount of time and money of the clients; therefore, helps in running their business smoothly.

#### 5. What is a Cloud Service?

- A cloud service is a service that is used to build cloud applications. This service provides the facility of using the cloud application without installing it on the computer. It reduces the maintenance and support of the application as compared to those applications that are not developed using the cloud service. The different kinds of users can use the application from the cloud service, which may be public or private application.

#### 6. What are main features of cloud services?

- Some important features of the cloud service are given as follows:
  - Accessing and managing the commercial software
  - Centralizing the activities of management of software in the Web environment
  - Developing applications that are capable of managing several clients
  - Centralizing the updating feature of software that eliminates the need of downloading the upgrades

### 7. What are the advantages of cloud services?

- Some of the advantages of cloud service are given as follows:
  - Helps in the utilization of investment in the corporate sector; and therefore, is cost saving.
  - Helps in the developing scalable and robust applications. Previously, the scaling took months, but now, scaling takes less time.
  - Helps in saving time in terms of deployment and maintenance.

### 8. Explain different models of cloud service.

- There are three distinct models in which the different services can be grouped. These three cloud service delivery models are:

*Infrastructure as a Service (IaaS) or Cloud infrastructure services*—Provides storage and compute resources that developers and IT organizations can use to offer customized business solutions.

The following are some of the advantages of IaaS:

- Delivers servers, networking technology, storage, and data center space as a service and also includes the delivery of operating systems and virtualization technology to manage the resources

- Includes dynamic scaling by which a user can request for more resources than expected

*Platform as a Service (PaaS)*—Enables the provider to deliver the additional amount of infrastructure. It offers a solution stack, which is a mutually inclusive set of software to help developers at the time of development of application and its execution. The example of PaaS is Force.com.

The following are some of the advantages of PaaS:

- Decreases the maintenance effort as the application is deployed on the cloud
- Supports all the Web service standards and often delivers with the dynamic scaling, which means that the software can extend and shrink its services
- **Software as a Service (SaaS)**—Delivers the software as a service over the Internet. SaaS also provides security and reliability by incorporating Virtual Private Networks (VPNs). This enables the user to transfer the data securely over the network, such as Internet. The example of SaaS is yahoo mail, which can handle millions of users.

### 10. Explain public and private cloud.

- **The public cloud** (or external cloud) is freely available for access. You can use a public cloud to collect data of the purchasing of items from a Web site on the Internet. You can also use public cloud for the reasons, which are given as follows:

- Helps when an application is to be used by a large number of people, such as an e-mail application, on the Internet
- Helps when you want to test the application and also needs to develop the application code
- Helps when you want to implement the security for the application
- Helps when you want to increase the computing capacity
- Helps when you are working on the projects in collaboration
- Helps when you are developing the project on an ad-hoc basis by using PaaS

**The private cloud** allows the usage of services by a single client on a private network. The benefits of this model are data security, corporate governance, and reliability concerns. The private cloud is used by the organization when it has a huge, well-run data center having a lot of spare capacity. It is also used when an organization is providing IT services to its clients and the data of organization is highly important. It is best suited when the requirements are critical.

The characteristics of this model are given as follows:

- Provides capability to internal users and allows provision of services
- Automates the tasks of management and provides the billing of consumption of a particular service

The following are some of the advantages of SaaS:

- Offers service on per-use basis and involves no hidden costs from the service provider
- Provides the flexibility to test new software on a rental basis and then on satisfaction can adopt the software

- Offers a well-managed environment
- Enables the optimization of computational resources, such as servers
- Manages the workload of the hardware
- Offers self-service based provisioning of hardware resources and software

#### 11. Explain hybrid and community cloud.

The **hybrid cloud** consists of multiple service providers. This model integrates various cloud services for Hybrid Web hosting. It is basically a combination of private and public cloud features. It is used by the company when a company has requirements for both the private and public clouds. Consider an example when an organization wants to implement the SaaS application throughout the company. The implementation requires security that can be provided by the private cloud used inside the firewall. The additional security can be provided by the VPN on requirement. Now, the organization has both the private and public cloud features.

The **community cloud** provides a number of benefits, such as privacy and security. This model, which is quite expensive, is used when the organizations having common goals and requirements are ready to share the benefits of the cloud service.

#### 12. Why does an organization need to manage the workloads?

The workload can be defined as an independent service or a set of code that can be executed. It can be everything from a data-intensive workload to storage or a transaction processing workload and does not rely upon the outside elements. The workload can be considered as a small or complete application.

The organization manages workloads because of the following reasons:

- To know how their applications are running
- To know what functions they are performing
- To know the charges of the individual department according to the use of the service

#### 13. Give a brief introduction of Windows Azure operating system.

The Windows Azure operating system is used for running cloud services on the Windows Azure platform, as it includes necessary features for hosting your services in the cloud. It also provides runtime environment that consists of Web server, computational

- services, basic storage, queues, management services, and load balancers. The operating system provides development fabric for development and testing of services before their deployment on the Windows Azure in the cloud.

#### 14. Which services are provided by Window Azure operating system?

- Windows Azure provides three core services which are given as follows:
  - Compute** —Offers scalable hosting of services on Windows Server 2008 operating system or platform. The platform has been designed to scale, based on the requirements of the user. This platform is based on the concept of virtualization. The platform executes on the Internet Information Server (IIS) 7 or higher version, which can be used with ASP.NET Web applications. The operating system layer also includes abstraction that means only important information should be available to the user instead of all the information. The developers uses the abstraction for developing managed and unmanaged services for deploying in the Windows Azure computing cloud without going in depth of the organization of the operating systems.
  - Storage** —Stores data in the form of tables, blobs, and queues and is supported by Windows Azure. The tables present in the Windows Azure are not similar to relational database tables, such as SQL Server tables. The Windows Azure has an entity data model, which is an independent data model. It provides structured data storage capabilities. The tables present in the database can store large amount of data, such as in terabytes. Windows Azure blobs have the capability to store large collection of binary data, such as videos, images, and music, in the cloud environment. A single blob can contain 50 GB of data. The queues present in the Windows Azure acts as asynchronous communication channels for connecting services between services and applications present in a cloud. In addition, they can be used to establish communication between multiple Windows Azure role instances and support unlimited number of messages; however, the size of the message cannot exceed 8KB.
  - Management** —Provides the automated infrastructure and capabilities of service management to Windows Azure cloud services. The main features of the management capabilities are automated virtual machines, deploying services in virtual machines, configuring switches, access routers, and load

balancers to maintain the user-defined state of the service. The fabric controller present in the management service maintains the health of the service and abstracts the infrastructure of virtual platform from the compute and storage services. Dynamic upgradeation of services is also supported by the fabric controller. The Windows Azure management service supports features, such as custom logging, tracing, and service usage monitoring.

### 15. What is the AppFabric component?

The **AppFabric** component is used to create access control and distribute messages across clouds and enterprises. It has a service-oriented architecture, and can be considered as the backbone of the Windows Azure platform. It provides connectivity and messaging among distributed applications. It also has the capabilities of integrating the applications and the business processes between cloud services and also between cloud services and global applications. The AppFabric component provides a development environment that is integrated with Visual Studio 2010. The Windows Communication Foundation (WCF) services built in VS 2010 can be published on cloud from the Visual Studio design environment.

The two important services of AppFabric are as follows:

- **Access Control Service (ACS)**—Allows rules-driven and claims-based access control for distributed applications. These claims-based rules and authorization roles can be defined in the cloud for accessing on-premise and cloud services. The claim can be a user or application attribute, which the service application expects, such as e-mail address, phone number, password, and role, for appropriate access control. When any application wants to use the Web service, it sends the required claims to ACS for requesting a token. ACS converts the input claims into output claims by following the rules of mapping. These rules are created during the configuration of ACS. The ACS issues a token containing the output claims for the consumer application. This application uses this token in the request header and sends to the Web service. This service validates the claims in the token and gives suitable access to the user.
- **Service bus**—Provides messaging between cross-enterprise and cross-cloud scenarios. It provides publish/subscribe, point-to-point, and queues message patterns for exchange of messages across distributed applications in the cloud. It integrates with the Access Control service to establish secure relay and communication.

## Question Index

- |  |  |
|--|--|
| <b>Chapter 1: .NET Framework</b>   |  |
| 1. What is .NET Framework? 1   |  |
| 2. Explain the different versions of .NET. 1   |  |
| 3. What are the main components of .NET Framework? 3   |  |
| 4. List the new features added in .NET Framework 4.0. 4  |  |
| 5. What is difference between System.String and System.StringBuilder classes? 5  |  |
| 6. Describe the roles of CLR in .NET Framework. 6  |  |
| 7. What is the role of the JIT compiler in .NET Framework? 6   |  |
| 8. What is Microsoft Intermediate Language (MSIL)? 7   |  |
| 9. What is Common Language Specification (CLS)? 7  |  |
| 10. What is Common Type System (CTS)? 7  |  |
| 11. Differentiate between managed and unmanaged code? 7  |  |
| 12. State the differences between the Dispose () and Finalize (). 8  |  |
| 13. Name the classes that are introduced in the System.Numerics namespace. 8   |  |
| 14. How do you instantiate a complex number? 9   |  |
| 15. What is lazy initialization? 9   |  |
| 16. Give a brief introduction on side-by-side execution. Can two applications, one using private assembly and the other using the shared assembly be stated as side-by-side executables? 9 |  |
| 17. Explain covariance and contravariance in .NET Framework 4.0. Give an example for each. 9   |  |
| 18. What is garbage collection? Explain the difference between garbage collections in .NET 4.0 and earlier versions. 10  |  |
| 19. How many types of generations are there in a garbage collector? 10   |  |
| 20. What are tuples? 11  |  |
| 21. How can you instantiate a tuple? 11  |  |
| 22. What are the improvements made in the file system enumeration in .NET 4.0? 11  |  |
| 23. Explain memory-mapped files. 12  |  |
| 24. What is the difference between int and int32. 12   |  |
| 25. What is code access security (CAS)? 12   |  |
| 26. How does CAS works? 12   |  |

27. How can you turn-on and turn-off CAS? 13  
 28. What are the improvements made in CAS in .NET 4.0? 13  
 29. Which method do you use to enforce garbage collection in .NET? 13  
 30. What are code contracts? 13  
 31. Which is the root namespace for fundamental types in .NET Framework? 14  
 32. What is managed extensibility framework? 14  
 33. Is there a way to suppress the finalize process inside the garbage collector forcibly in .NET? 14  
 34. Mention the execution process for managed code. 14
- Chapter 2: Programming Concepts**
- Define variable and constant. 15
  - What is a data type? How many types of data types are there in .NET. 15
  - Mention the two major categories that distinctly classify the variables of C# programs. 16
  - Which statement is used to replace multiple if-else statements in code. 16
  - What is the syntax to declare a namespace in .NET? 16
  - Determine the output of following code snippet: 16
  - Briefly explain the characteristics of value-type variables that are supported in the C# programming language. 17
  - Briefly explain the characteristics of reference-type variables that are supported in the C# programming language. 17
  - Differentiate between Boxing and Unboxing. 17
  - Mention different types of comments that can be inserted in a code. 17
  - What is an identifier? 18
  - What is the difference between constants and read-only variables that are used in programs? 19
  - Explain keywords with example. 19
  - What are the different types of literals? 19
  - Give the syntax of using the for loop in C# code? 20
  - Give the syntax of using the while loop in a C# program. 21
  - Differentiate between the while and for loop in C#. 21
  - What are the different iteration statements in Visual Basic? 21
  - Identify the correct method to declare an array in VB.NET from the following options: 23
  - List all the types of access modifiers. 23

21. What is the main difference between sub-procedure and function? 24  
 22. What does a break statement do in the switch statement? 25  
 23. Write a program to swap two numbers and print them in C#. 25  
 24. What is a parameter? Explain the new types of parameters introduced in C# 4.0. 26
- Chapter 3: Object-Oriented Programming**
- What is object-oriented programming (OOP)? 27
  - What is a class? 27
  - What is an object? 27
  - What is the relationship between a class and an object? 27
  - Explain the basic features of OOPS. 28
  - What are abstract classes? What are the distinct characteristics of an abstract class? 28
  - When do you really need to create an abstract class? 28
  - State the features of an interface. 29
  - Differentiate between an abstract class and an interface. 29
  - What are methods? 30
  - Give a brief description of properties in C# and the advantages that are obtained by using them in programs. 30
  - What do you mean by data encapsulation? 30
  - Explain different types of inheritance. 30
  - Explain the concept of constructor? 31
  - Explain the concept of destructor? 31
  - What is a static constructor? 31
  - You have defined a destructor in a class that you have developed by using the C# programming language, but the destructor never executed. Why did the destructor not execute? 32
  - Why is the virtual keyword used in code? 32
  - Does .NET support multiple inheritance? 32
  - How can you prevent your class to be inherited further? 32
  - A structure in C# can implement one or more interfaces. Is it true or false? 32
  - What is a delegate? 32
  - Define an event. 33
  - What is a multicast delegate? 33
  - Do events have return type? 33

26. How can you prevent a class from overriding in C# and Visual Basic? 33
27. What is the syntax to inherit from a class in C#? 33
28. Can you allow a class to be inherited, but prevent a method from being overridden in C#? 33
29. How is method overriding different from method overloading? 33
30. Can you declare an overridden method to be static if the original method is not static? 33
31. What are the different ways a method can be overloaded? 34
32. Is it possible for a class to inherit the constructor of its base class? 34
33. What is the difference between procedural and object-oriented programming? 34
34. Can you specify the accessibility modifier for methods inside the interface? 34
35. Can you declare a private class in a namespace? 34
36. Can you inherit private members of a class? 34
37. What is a namespace? 34
38. What is the difference between a class and a structure? 35
39. What are similarities between a class and a structure. 35
40. Define an array. 35
41. What is the index value of the first element in an array? 36
42. In which namespace, all .NET collection classes are contained? 36
43. What are collections and generics? 36
44. What are queues and stacks? 36
45. What are structures? 36
46. What is the difference between arrays and collection? 37
47. Define enumeration? 37
48. What is a hashtable? 37
49. What is the function of the Try-Catch-Finally block? 37
50. Is it a good practice to handle exceptions in code? 38
51. Is it possible to execute two catch blocks? 38
52. Can users define their own exceptions in code? 38
53. Can you use the throws clause to raise an exception? 38
54. How has exception handling changed in .NET Framework 4.0? 38

**Chapter 4: Windows Control**

- How can we auto size a button to fit its text? 39
- Explain the method adopted to customize the visual appearance of the Button control in C#. 39

- How can we display an icon or a bitmap image on the Button control? 40
- Which method is used to generate the click event of the Control class for the Button control in C#? 40
- A Windows Form will not show the Minimize, Maximize, and Close buttons, if the ControlBox property of the form is set to False. (True/False) 40
- How is anchoring different from docking? 40
- How can we disable the context menu for a TextBox control? 40
- How do we format numbers, dates, and currencies in a text box? 40
- Write the code that enables to browse for a text file in a system and get its content in a text box control in C#. 40
- How can you enforce a TextBox to display characters in uppercase? 41
- How can you enable a TextBox to change its characters format, so that users can enter password? 41
- How can you enable a read-only text box to ignore mouse down events so that users can scroll the text and position the cursor in .NET 4.0? 42
- Which event of a TextBox control helps in restricting a TextBox from accepting numeric digits in .NET 4.0? 42
- When you set a TextBox to ReadOnly, the text color is gray. How can you change this color to the one specified in the ForeColor property of the TextBox in .NET 4.0? 42
- Write the code to shift the focus from a TextBox to another control on a form when a user presses the ENTER key? 43
- Write the code to create and save a Rich Text Format (RTF) file in C#. 44
- How can you get the text of the RichTextBox control, including all rich text format strings in .Net 4.0? 44
- Write the steps necessary to enable a RichTextBox control to support drag and drop in .Net 4.0? 44
- How can you set the color and font for the text displayed in the RichTextBox control in .NET 4.0? 45
- How can you programmatically position the cursor on a given line or on a character in the RichTextBox control in C#? 46
- How can you add a hyperlink to a RichTextBox control in .NET 4.0? 46
- How can you place a border around a picture box? 47
- How can you copy a bitmap from the clipboard to a picture box in .NET 4.0? 47

24. How can you drag and drop an image from one picture box to another in .NET 4.0? 47
25. How can you programmatically prevent a Combobox from dropping, in .NET 4.0? 49
26. How can you prevent users of an application from editing the text in the ComboBox controls in .NET 4.0? 49
27. How can you adjust the height of a combo box drop-down list? 49
28. How do you set the width of a Combobox to fit the entries in its list in C#? 49
29. How can you programmatically create a new list for the combo box drop-down list in .NET 4.0? 50
30. How can you drag file names from Windows Explorer and drop them in a list box in .NET 4.0? 50
31. Write the code to set the width of a list box to fit the text in C#. 51
32. How can you check/uncheck all items in the CheckedListBox control in .NET 4.0? 52
33. How will you pick a color from the ColorDialog box? 52
34. How do you use the FontDialog class to set a control's font in C#? 52
35. Write a method to get only the name of a file from the complete path string in C#. 53
36. How do you add subitems to a ListView control in C#? 53
37. How can you unselect the selected items in a ListView control programmatically in .NET 4.0? 53
38. Write a code to select an item in the ListView control programmatically in C#. 53
39. How can we check whether a child form is already displayed or not, so that we do not have two instances of a child form? 53
40. We need to perform certain custom processing whenever an MDI child form is added /removed from the MDIContainer form. How do we achieve this in C#? 54
41. How can you make your child form fill the entire MDI client without getting maximized in .NET 4.0? 55
42. How does an MDI form differ from a standard form? 55
43. Can you write a class without specifying a namespace? Which namespace does it belong to by default? 55
44. What are the three states set in the CheckState property of a Checkbox? 56
45. What is the difference between the Add() and Insert() methods of a ListView control? 56

46. Where does an ImageList control appear when you add it at the design time? 56
47. Consider a situation where you have added panels in a StatusBar control; however, they are not displayed at run time. What could be the reason for this? 56
48. Does a Timer control appear at run time? 56
49. What is the difference between a MenuStrip control and a ContextMenuStrip control? 56
50. What is the difference between a toolStrip drop-down button and a toolStrip split button? 56
51. What is the use of a toolStrip container? 56
52. Is it possible to associate a control with more than one ContextMenu? 57
53. What is the difference between a CheckBox control and a RadioButton control? 57
54. Why do you require user-defined controls? 57
55. What is the difference between a ListBox control and a ComboBox control? 57
56. What is the difference between the Panel and GroupBox control? 57
57. What is the importance of a Button control? 57
58. What are the values that can be assigned to the DialogResult property of a Button control? 57
59. How can you move and resize a control on a Windows form? 58
60. When would you use the ErrorProvider control? Provide an example of using the ErrorProvider control in C#. 58
61. How would you create an ellipse, which is a non-rectangular window? 58
62. How do you create a separator in the Menu Designer? 59
63. What is the difference between the WindowsDefaultLocation and WindowsDefaultBounds properties? 59
64. What is the difference between pixels, points, and em's when fonts are displayed? 59
65. How do you retrieve the customized properties of a .NET application from the XML .config file? 59
66. How can you display a default value in the text box of an input box? 59
67. Name the parent class for all Windows controls. 59
68. How can you add a Button control dynamically at runtime in C#? 59

69. What does the DialogResult property of a Button control do? 60
70. Is it possible to enter more than one line in a Textbox control? 60
71. Differentiate between a Textbox control and Rich Textbox control. 60
72. What is the MaskedTextBox control? What does the Mask property do? 61
73. Is it possible to add an image on the RadioButton control? 61
74. Create an application in C# to accept user name and password. The application must check the user name and password whether it is valid or not. Application will provide three login attempts to a user. If the user does not succeed to login, then the application should terminate. 61
75. What is the function of the CheckState property of the CheckBox control? 62
76. How can you attach a horizontal scroll bar with the ListBox control? 62
77. Name the methods, available in .NET 4.0, that are used to add and delete items from a ListBox control? 62
78. Can you add more than one item simultaneously in the ListBox control? 62
79. What is use of the DropDownStyle property of the ComboBox control? 62
80. What is the difference between FindString() and FindStringExact() methods of the Combobox control? 62
81. Which of the following controls do not have the Items property? 63
82. What is the function of theSizeMode property of the PictureBox control? 63
83. What is the use of the Panel control? Does it display at runtime? 63
84. How can you add a scroll to a Panel control? 63
85. How can you add a new tab page in a Tab control in C#? 64
86. Define the TrackBar control. 64
87. What does the TickFrequency property of the TrackBar control do? 64
88. Describe the ToolTip control. How can you associate it with other controls? 64
89. What is the function of MinDate and MaxDate properties of the MonthCalendar control? 64
90. What does the Checked property of the DateTimePicker control do? 64

91. What is the use of a Timer control? Can a Timer control pause? 65
92. How can you get or set the time between Timer ticks? 65
93. What does the PerformStep() method do? 65
94. Which class manages the event and layout of all ToolStrip elements? 65
95. Name the classes used to handle standard menu in a ToolStrip control. 65
96. Is it possible to associate a control with more than one ContextMenu control? 65
97. What is a StatusStrip control? 65
98. How can you display an icon at runtime on the StatusStrip control? 66
99. Which method provides the functionality to display a dialog box at runtime? 66
100. What does the OpenFileDialog() method of the OpenFileDialog control do? 66
101. What is a PrintPreview control? Give the function of the Document property of the PrintPreview control? 66
102. How many parameters are required to the ErrorProvider.SetError() method? 66
- Chapter 5: ADO.NET**
- Mention the full form of ADO. 67
  - Explain ADO.NET in brief. 67
  - What are the benefits of using of ADO.NET in .NET 4.0. 67
  - Explain the architecture of ADO.NET in brief. 68
  - Mention different types of data providers available in .NET Framework. 69
  - Explain in brief DataAdapter class in ADO.NET. 69
  - What is a DataReader object? 69
  - What is the role of the DataSet object in ADO.NET? 70
  - Describe the disconnected architecture of ADO.NET's data access model. 70
  - Mention the namespace that is used to include .NET Data Provider for SQL server in .NET code. 70
  - What are the usages of the Command object in ADO.NET? 70
  - What are the various methods provided by the DataSet object to generate XML? 71
  - How can you identify whether or not any changes are made to the DataSet object since it was last loaded? 71

14. How can you add or remove rows from the DataTable object of DataSet? 71
15. What is the use of DataView? 72
16. What is the use of the CommandBuilder class? 72
17. What is the meaning of object pooling? 72
18. What is the use of the connection object? 72
19. What is connection pooling? 73
20. What are the parameters that control most of connection pooling behaviors? 73
21. What are the pre-requisites for connection pooling? 73
22. Explain the DataAdapter.Update() and DataSet.AcceptChanges() methods. 73
23. What property must be set and what method must be called in your code to bind the data from some data source to the Repeater control? 73
24. Name the method that needs to be invoked on the DataAdapter control to fill the generated DataSet with data? 74
25. Name the two properties of the DataGridView control that have to be specified to turn on sorting and paging. 74
26. Which namespaces are required to enable the use of databases in ASP.NET pages? 74
27. Which object is used to add a relationship between two DataTable objects? 74
28. What are different types of authentication techniques that are used in connection strings to connect .NET applications with Microsoft SQL Server? 74
29. Out of Windows authentication and SQL Server authentication, which authentication technique is considered as a trusted authentication method? 75
30. How can you implement transactions in ADO.NET? 75
31. How would you connect to a database by using .NET? 75
32. What is the difference between OLEDB Provider and SqlConnection? 75
33. Which property is used to check whether a DataReader is closed or opened? 76
34. What is the difference between the Clone() and Copy() methods of the DataSet class? 76
35. Which architecture does Datasets follow? 76
36. Which adapter should you use, if you want to get the data from an Access database? 76
37. Which properties are used to bind a DataGridView control? 76

38. Explain about ADO.NET Entity Framework in brief. 76
39. Explain the new features in ADO.NET Entity Framework 4.0. 78
40. 78

### **Chapter 6: Language-Integrated Query**

1. What is Language Integrated Query (LINQ)? 79
2. What are LINQ query expressions? 79
3. Write the basic steps to execute a LINQ query. 79
4. Write the basic syntax of a LINQ query in Visual Basic as well as in C#. 79
5. In which statement the LINQ query is executed? 80
6. What are standard query operators in LINQ? 80
7. Which interface implements the standard query operators in LINQ? 80
8. List the various standard query operators based on their functionality. 80
9. What is the function of the Distinct clause in a LINQ query? 81
10. Which among these is used as a filtering operator? 81
11. What is the difference between the Take and Skip clauses? 81
12. What type of join does the Join clause perform? 81
13. On what parameter does the GroupBy clause group the data? 82
14. What is the difference between the Select clause and SelectMany() method in LINQ? 82
15. What are the different implementations of LINQ? 82
16. What is Object Relational Designer (O/R Designer)? 82
17. How can you open the O/R Designer? 82
18. What is the DataContext class and how is it related to LINQ? 83
19. Before you query a DataSet object by using LINQ to DataSet, you must first populate the dataset. How can you do this? 83
20. What are the different Visual Basic features that support LINQ? 83
21. The standard query operators are themselves a set of extension methods that provide the LINQ query functionality for any type that implements the IEnumerable<T> interface in Visual Basic. Is it True or False? 83
22. What are lambda expressions in LINQ? 83
23. In LINQ, lambda expressions underlie many of the standard query operators. Is it True or False? 84

24. Name the control that exposes the LINQ features to Web developers through the ASP.NET data-source control architecture. 84
25. What is a LinqDataSource control? 84
26. Which command-line tool generates code and mapping for the LINQ to SQL component of .NET Framework? 84
27. Write the name of the property that you need to connect the LinqDataSource control to a database class. 84
28. Which is the new operator added in LINQ? 85
29. Write a note on the Zip operator in .NET Framework 4.0. 85
30. What is PLINQ? 85
31. Which extension method do you need to run a parallel query in PLINQ? 86

**Chapter 7: Dynamic Programming**

1. What is Dynamic Language Runtime (DLR)? 87
2. What are the advantages of DLR? 87
3. Give a brief introduction to Binders. 87
4. Explain the different services provided by DLR to CLR. 88
5. Name the binders provided by .NET Framework 4.0. 88
6. What is Dynamic Data Type? 89
7. What is the difference between dynamic and var data types? 89
8. Which class is used for converting the data types? 89
9. Explain ExpandoObject and DynamicObject classes. 89

**Chapter 8: Extensible Markup Language (XML)**

1. Write a note on Extensible Markup Language (XML). 91
2. Explain the difference between XML and HTML. 91
3. Describe the rules and regulations that must be followed while creating a well-formed XML document. 91
4. What is an XML attribute? 92
5. Explain the XML elements. 92
6. What are the naming conventions required for XML elements tags? 92
7. What is the DTD? 93
8. What is an XML schema? 93
9. State the advantages of XML schemas over DTD. 93
10. What is an XML namespace? 93
11. Which namespaces in .NET are used for XML? 94
12. The XML preserves white spaces. Is it true? 94

**Chapter 9: ASP.NET**

1. What is ASP? 99
2. What is ASP.NET? 99
3. What is the basic difference between ASP and ASP.NET? 99
4. What is an ASP.NET Web Form? 100
5. Explain the new features of ASP.NET 4.0 in brief. 100
6. What is IIS? Why is it used? 102
7. Among the following, which is the latest version of IIS? 102
8. What are the events that happen when a client requests an ASP.NET page from IIS server? 102

13. The XML elements cannot be empty. Is it true? 94
14. Is it true that the XML's goal is to replace HTML? 94
15. Out of the following four options, which one is the correct syntax to define the XML version in a XML document? 94
16. Is the following document a well-formed XML document? 95
17. Is the following document a well-formed XML document? 95
18. Which of the following is a correct name for an XML element? 95
19. Which is the correct way of referring to a style sheet (styles.xsl)? 95
20. Which are the five special characters that cannot be included in an XML document? 95
21. What is XSLT? 96
22. Using XSLT, how would you extract the value of a specific attribute from an element in an XML document? 96
23. Describe the role that XSL can play while dynamically generating HTML pages from a relational database. 96
24. What is XPath? 96
25. What is XML DOM? 96
26. Give an example of a DOM-enabled XML parser. 97
27. Explain different types of XML parsers. 97
28. What are the advantages of DOM? 97
29. Which classes are supported to make an XML DOM? 97
30. Which class is used to encode and decode XML names and contains different methods to convert between CLR types and XSD types. 97
31. Name the W3C standards supported by .NET Framework. 98
32. Explain the XmlReader class. 98
33. Describe the XmlWriter class. 98
34. What is the function of the XmlDocument class? 98

9. How would ASP and ASP.NET applications run at the same time on the same server? 102  
Or
- How does IIS handle an ASP and ASP.NET request? 102
10. What is the process of the server-side application architecture? 103
11. What is the process of the client-side application architecture? 103
12. What is actually returned from server to the browser when a browser requests an .aspx file and the file is displayed? 104
13. What events are fired when a page loads? 104
14. Which event determines that all the controls are completely loaded into memory? 104
15. To which class a Web form belongs to in the .NET Framework class hierarchy? 104
16. How information about the user's locale can be accessed? 105
17. What is a round trip? 105
18. How can you identify that the page is PostBack? 105
19. What are the HTML server controls in ASP.NET? 105
20. What are Web server controls in ASP.NET? 105
21. What are Custom User Controls in ASP.NET? 105
22. What is the difference between Web User Control and Web Custom Control in ASP.NET? 105
23. What are the major built-in objects in ASP.NET? 106
24. What are HTTP handlers in ASP.NET? 106
25. Explain the concept of states in ASP.NET. 107
26. Explain the Application and Session objects in ASP.NET. 107
27. What is State Management? How many ways are there to maintain a state in .NET? 108
28. What is the code-behind feature in ASP.NET? 108
29. What are the advantages of the code-behind feature? 108
30. Suppose you want an ASP.NET function (client side) executed on the MouseOver event of a button. Where do you add an event handler? 109
31. Which ASP.NET objects encapsulate the state of the client and the browser? 109
32. What is the behavior of a Web browser when it receives an invalid element? 109
33. Which class is inherited when an ASP.NET server control is added to a Web form? 109

34. Explain the validation controls. How many validation controls in ASP.NET 4.0? 109
35. How can you display all validation messages in one control? 109
36. Which data type does the RangeValidator control support? 110
37. Which control will you use to ensure that the values in two different controls match? 110
38. Can you post and access view state in another application? 110
39. What is the function of the CustomValidator control? 110
40. What happens if an ASP.NET server control with event-handling routines is missing from its definition? 110
41. Which method is used to force all the validation controls to run? 110
42. Where is the ViewState information stored? 110
43. What does the "EnableViewState" property do? Why do we want it On or Off? 110
44. What is the use of web.config? What is the difference between machine.config and web.config? 110
45. What are the various ways of authentication techniques in ASP.NET? 111
46. What is Role-based security? 111
47. What is the appSetting Section in the web.config file? 112
48. What is the use of the <sessionState> tag in the web.config file? 112
49. What is Query String? What are its advantages and limitations? 112
50. What is a Cookie? Where is it used in ASP.NET? 112
51. How many types of Cookies are available in ASP.NET? 113
52. What is the default timeout for a Cookie? 113
53. Explain how Cookies work. Give an example of Cookie abuse. 113
54. Which method do you use to kill explicitly a users session? 113
55. What are the different ways to send data across pages in ASP.NET? 113
56. How do you sign out from forms authentication? 113
57. What is the use of the Global.asax file? 113
58. What are the event handlers that can be included in the Global.asax file? 113

59. Which namespaces are necessary to create a localized application? 114
60. What is tracing? Where is it used? 114
61. Which ASP.NET configuration options are supported in the ASP.NET implementation on the shared Web hosting platform? 114
62. What are server-side comments? 115
63. How can you dynamically add user controls to a page? 115
64. What is the difference between authentication and authorization? 115
65. Is there a way to set Option Explicit and Option Strict in an ASP.NET page without using a code-behind? 115
66. How can you upload a file from an ASP.NET page? 115
67. How can you send an email message from an ASP.NET Web page? 116
68. Which page event checks the IsPostBack property during the page life cycle? 116
69. Which compilation model is best for highly secure Web sites? 116
70. Which directory contains the code for the utility classes? 116
71. Which page directive is only used in the user controls? 116
72. What is the difference between HTML and Web server controls? 116
73. What type of code, client-side or server-side, is found in a code-behind file of a Web page? 116
74. After capturing the SelectedIndexChanged event for a DropDownList control, you find that the event handler does not execute. What can be the problem? 117
75. What is the difference between a HyperLink control and a LinkButton control? 117
76. What is the difference between an HtmlInputCheckBox control and an HtmlInputRadioButton control? 117
77. Why a SiteMapPath control is referred to as breadcrumb or eyebrow navigation control? 117
78. Which of the following property allows you to skip intermediate sequential steps in a Wizard control? 117
79. What does the Orientation property do in a Menu control? 117
80. Unlike the Menu or Treeview control, the SiteMapPath control does not use a SiteMapDataSource control. Instead, it relies on a SiteMapProvider to retrieve the data that it will display. Is it true or false? 118
81. Differentiate between client-side and server-side validations in Web pages. 118
82. Where should the data validations be performed—at the client side or at the server side and why? 118
83. How do you prevent a validation control from validating data at the client end? 118
84. How can you check if all the validation controls on a Web page are valid and proper? 118
85. What does the WebpartListUserControlPath property of a DeclarativeCatalogPart control do? 118
86. What does the .WebPart file do? 118
87. In which database is the information, such as membership, role management, profile, and Web parts personalization, stored? 119
88. What type of the CatalogPart control enables users to restore the Web Parts that have been removed earlier by the user? 119
89. How does a content page differ from a master page? 119
90. Why do we need nested master pages in a Web site? 119
91. How will you differentiate a submaster page from a top-level master page? 119
92. Where is a skin file stored in a web application? 119
93. What is the difference between a page theme and a global theme? 119
94. What is the difference between a default skin and a named skin? 119
95. Which of the following values does the <passwordFormat> attribute take? 120
96. What setting must be added in the configuration file to deny a particular user from accessing the secured resources? 120
97. How can you enable impersonation in the web.config file? 120
98. Define a multilingual Web site. 120
99. Differentiate globalization and localization. 120
100. What do you understand by the culture? 120
101. What do you mean by a neutral culture? 120
102. What are the Culture and UICulture values? 121
103. What is the difference between page-level caching and fragment caching? 121
104. Suppose a Web page has been cached for 60 seconds and a user control added to this Web page has been cached for 30 seconds. After how much time, the cache for the user control expires. 121

105. What is the difference between absolute expiration and sliding-time expiration? 121
106. If we have cached a dataset that has a sliding time expiration set as 20 seconds and we access this dataset in 18th second after being cached, the total time for which the dataset will be present in the cache is: 121
107. Explain file-based dependency and key-based dependency. 121
108. What do you understand by aggregate dependency? 121
109. What is the difference between adding items into cache through the Add() method and through the Insert() method? 122
110. What is the difference between SQL notification and SQL invalidation? 122
111. What is the difference between ASP session and ASP.NET session? 122
112. Explain the cookie less session and its working. 122
113. How can you implement the postback property of an ASP.NET control? 122
114. What is AutoPostBack? 122
115. How can you ensure that no one has tampered with ViewState in a Web page? 122
116. How can you register a custom server control to a Web page? 123
117. How can you assign page specific attributes in an ASP.NET application? 123
118. Which method is used to post a Web page to another Web page? 123
119. What is cross-page posting in ASP.NET? 123
120. Why do you use the App\_Code folder in ASP.NET? 123
121. What do you mean by the Web Part controls in ASP.NET? 123
122. Write a code snippet to start a Notepad file in ASP.NET. 123
123. What is ViewState? 124
124. Describe the complete lifecycle of a Web page. 124
125. What is the difference between the Response.Write() and Response.Output.Write() methods? 124
126. What is the lifespan for items stored in ViewState? 124
127. Which of the following data bind controls does not support the sorting feature? 124
128. Make a list of all templates of the Repeater control. 125
129. Which of the following templates cannot be skipped in the Repeater control? 125
130. Write three common properties of all validation controls. 125

131. Can we validate a DropDownList by RequiredFieldValidator? 125
132. Explain login controls. 125
133. What are navigation controls? How many navigation controls are there in ASP.NET 4.0? 126
134. Can you set which type of comparison you want to perform by the CompareValidator control? 126
135. How can we provide the WebParts control functionality to a server control? 126
136. Which control is responsible to create a space on Web pages? 126
137. Which is the parent class of the Web server control? 126
138. What is difference between a Label control and a Literal control? 127
139. What is the use of PlaceHolder control? Can we see it at runtime? 127
140. Explain the AdRotator Control. 127
141. Which method has been introduced in ASP.NET 4.0 to redirect a page permanently? 127
142. Which two new properties are added in ASP.NET 4.0 Page class? 127
143. What is the function of the ViewState property? 127
144. Which are two new controls introduced in ASP.NET 4.0. Explain them. 128
145. List the features of the Chart control. 128

#### **Chapter 10: Windows and Web Services**

1. What are Windows services? 129
2. Can you share a process between Windows services? 129
3. In .NET, which is the parent class to create all Windows services? 129
4. Which class in .NET is used to install a Windows service? 129
5. While installing a Windows service, an EventLogInstaller class is automatically created to install the event log related to the particular service. Is it true? 129
6. Which tool installs the DLL on your local computer and installs the Windows service in a transactional manner? 130
7. Which property of the ServiceBase class can be used to specify whether a service can be paused and resumed? 130
8. Which method is used to uninstall the Windows services? 130
9. Describe the EventLog class. 130
10. Explain the concept of Web services in brief. 130

**.NET INTERVIEW QUESTIONS**

11. What advantages have Web services over Component Object Model (COM) and Distributed Component Object Model (DCOM)? 131
12. Write the file extension for a Web service. 131
13. Which two methods are used to discover the URL of Web services? 131
14. How do Web services communicate with applications across diverse platforms and hardware? 131
15. What is the structure of a SOAP message that is used by Web services for communication? 131
16. What do you understand by SOAP encoding? 132
17. What is the use of a .disco file? 132
18. Which namespace must be included in a code that enables a XML Web service to write events in an event log file? 132
19. What is Universal Description, Discovery, and Integration (UDDI)? 132
20. Describe the services that UDDI provides to Web applications. 133
21. Explain the WSDL. 133
22. Which step is necessary to perform before a Web service can be consumed? 134
23. Write the names of public properties defined in the WebService class. 134
24. Mention the name of the directory where it is necessary to locate the proxy file to use a Web service. 134
25. Mention the namespace that you must import in code to build a Web service. 134
26. Give a description of the infrastructure and the Web resources that a client application uses to consume the services provided by a Web service. 134
27. How does a client application communicate with an XML Web service? 135
28. Which attribute is used to create the Web method? 137
29. What are the Page directives of a Web service's code? 137
30. Which property of the WebMethod attribute allows you to maintain the state of objects across sessions in a Web method? 138
31. What does a Session property do? 138
32. What steps do you need to perform to enable transactions in a Web method? 138
33. How can you ensure that only authorized users access your Web service? 138

***Chapter 11: Windows Workflow Foundation***

1. What is Windows Workflow Foundation (WF)? 141
2. What are the components of WF 4.0? 141
3. What is a workflow? 141
4. What are the different types of workflow in WF? 141
5. What are the four workflow principles? 142
6. What is an activity? 142
7. What is the difference between a system workflow and a human workflow? 142
8. What is a host process? 143
9. What is a runtime engine? 143
10. What are runtime services? 143
11. Explain the mygma1|passwordactivity data model of WF 4.0. 144
12. What are XOML files? 144
13. What is a dynamic update? 144
14. What is a base activity library? 144
15. Explain Custom Activities. 144
16. Explain why workflows are based on Extensible Models. 144
17. What are the new features in WF 4.0? 145
18. Explain the concept of Bookmarks in WF 4.0. 145
19. Which component of WF architecture is responsible to execute each workflow instance? 146
20. Write the steps that are involved in the sequential workflow, by default. 146
21. How can you implement a condition in a workflow? 146
22. What is the function of the Rule Condition Editor dialog box in WF? 146
23. Which option do you need to select for the Condition property, if you want to create a code condition? 146

24. Can you integrate workflow applications with some other application, such as Windows Forms applications and Web applications? 146

### **Chapter 12: Windows Presentation Foundation and Silverlight**

1. What is the main advantage of using Windows Presentation Foundation (WPF) over Windows Forms? 147
2. When was WPF introduced and what is its latest version? 147
3. Explain the architecture of WPF 4.0. 147
4. What are managed and unmanaged components in WPF 4.0? 148
5. Discuss in brief the architecture and programming model of Windows Presentation Framework (WPF). Also, mention the advantages of using both managed and unmanaged execution in WPF architecture. 148
6. What is the milcore? 149
7. Explain the WPF features. 149
8. What are the new improvements introduced in WPF 4.0? 150
9. Name some of the important classes of WPF 4.0. 150
10. What are the different types of applications that you can create in WPF using Visual Studio 2010? 151
11. Explain the main elements of WPF Designer. 151
12. What is XAML? Explain the use of XAML in WPF applications. Are there any advantages of incorporating XAML in WPF applications? 152
13. What is the extension of XAML files? 152
14. What are markup extensions and what are its benefits? Write the names of some common markup extensions available in WPF. 152
15. How is the property system of WPF different from the traditional CLR property system? What are dependency properties? 153
16. What is an attached property? Give an example. 153
17. What are routed events? What is the difference between bubbling and tunneling events? 154
18. What is a Binary Application Markup Language (BAML) file? 154
19. Which of the following is the new control introduced in WPF 4.0? 155
20. What are Container controls? 155
21. Define resources in WPF. What are the different types of resources? 155

22. What is the difference between Simple controls and Content controls? 156
23. What is the function of the BlackOutDates property of the DatePicker control? 156
24. What does a control template do? 156
25. What role does a Data template play in WPF controls? 156
26. What are triggers? 156
27. What do you understand by WPF commands? 157
28. Which classes are responsible for integrating WPF and Windows Forms technologies? 157
29. Name the classes that allow you to perform transformation for typography and text. 157
30. Which namespace allows you to work with documents in WPF? 158
31. Explain the types of documents supported by WPF. 158
32. Write the name of the control that allows you to work with fixed and flow documents. 158
33. What do you understand by serialization and deserialization of documents? 158
34. Mention the important classes in WPF that allow you to work with the 2-D graphics. 159
35. What are the brushes that are available in WPF? 159
36. How many types of projection cameras are available for 3-D graphics in WPF? 159
37. What are the different kinds of animation supported in WPF? 160
38. What are the image formats supported in WPF? 160
39. What are the formats of audio-video data that WPF supports? 160
40. What are the two modes of media playback? 161
41. What is the technique that is used by WPF applications to provide Help to the users? How it is different to the Help, which Windows controls provide to their users through pop-ups or online help? 161
42. What is the current version of Silverlight, which is supported by .NET Framework 4.0? 161
43. What is Silverlight? 161
44. Write the names of Web browsers that support Silverlight. 161
45. What are the tools or software packages available in the market that you can use to work with Silverlight? 162
46. What is an application package in Silverlight? 162
47. What are the primary languages that Silverlight uses? 162

**.NET INTERVIEW QUESTIONS**

48. Define the Deep Zoom technology in Silverlight. 162  
 49. What are the networking services that are supported by Silverlight? 162  
 50. What is a test page in Silverlight and when is it created? 162  
 51. Write the name of the control that can be used to enable the users to adjust the space in a grid. 162  
 52. Is it possible to work without adding the ScriptManager control in a Silverlight-enabled ASP.NET Web page? 163  
 53. What is the use of the MediaPlayer server control in ASP.NET? 163  
 54. What are the file formats supported by the MediaPlayer server control in ASP.NET? 163  
 55. What is the functionality provided by the Silverlight server control in ASP.NET? 163  
 56. Demonstrate how the Popup control in Silverlight is used with the help of an example. 163  
 57. What is the difference between Silverlight applications and XBAP applications in WPF? 164

**Chapter 13: Windows Communication Foundation 4.0**

1. What is Windows Communication Foundation (WCF)? 165  
 2. What are the benefits that WCF offers? 165  
 3. What was the code name of WCF? 166  
 4. When was WCF introduced? 166  
 5. Mention the different layers of WCF. 166  
 6. What is an endpoint? Briefly explain the components of a WCF endpoint. 167  
 7. Discuss the bindings that are available for WCF applications. 167  
 8. What are the different types of contracts available in WCF? 168  
 9. Briefly explain the basic steps required to create a service contract in WCF. 169  
 10. Discuss the message patterns supported by a service contract in WCF. 169  
 11. Explain the instancing model of WCF. 170  
 12. What does the Svcutil.exe tool perform? 170  
 13. Which protocol does WCF employ for distributed applications? 171  
 14. What do you understand by service identity? 171

**Chapter 14: ASP.NET AJAX**

1. What is ASP.NET AJAX? 175  
 2. What do you mean by the term asynchronous postback? 175  
 3. What is the difference between synchronous postback and asynchronous postback? 176  
 4. What technologies are being used in AJAX? 176  
 5. Why do we use the XMLHttpRequest object in AJAX? 176  
 6. How can we get the state of the requested process? 176  
 7. List the different states of XMLHttpRequest with their description. 176  
 8. What are the different ways to pass parameters to the server? 177  
 9. Is there any difference between HTML and XHTML? 177  
 10. Briefly describe ASP.NET AJAX Framework. 177  
 11. Explain the limitations of AJAX. 178  
 12. What are the differences between AJAX and JavaScript? 178  
 13. What is the importance of client-side libraries? 178  
 14. What are the components of the ASP.NET AJAX architecture? 178  
 15. What are the different controls of ASP.NET AJAX? 178  
 16. What is the use of the ScriptManager control in AJAX? 179  
 17. Is it possible to use multiple ScriptManager controls on a Web page? 179  
 18. What is the role of the ScriptManagerProxy control? 179  
 19. Describe the situations in which AJAX should not be used. 179  
 20. How can you find out that an AJAX request has been completed? 179  
 21. What is the syntax to create AJAX objects? 180  
 22. What is JSON? 180  
 23. Explain the UpdatePanel control. 180  
 24. Why do we use the UpdateProgress control in AJAX? 180  
 25. Explain the need of the Timer control in AJAX. 180  
 26. Can we call server-side code from JavaScript? 181

27. Can we nest the UpdatePanel controls? 181
28. Describe AJAX Control Extender Toolkit. 181
29. Is the AjaxControlToolkit.dll file installed in the Global Assembly Cache? 181
30. What are the requirements to run ASP.NET AJAX applications on a server? 181
31. What are the new features of ASP.NET AJAX 4.0? 181
32. Explain the live data-binding feature of ASP.NET AJAX 4.0. 182
33. What are the new features included in the Microsoft AJAX library? 182
34. What are the extender controls? 183
35. What are the new controls introduced in ASP.NET AJAX Control Toolkit? 183
36. How many validation controls are available in ASP.NET AJAX 4.0? 183
37. Describe the AccordionExtender control. 184
38. What is the work of the ConformOnFormSubmit property in the ConfirmButtonExtender control? 184
39. What does the DynamicPopulateExtender control do? 184
40. What does the MinimumPrefixLength property of the AutoCompleteExtender control do? 184
41. Explain the Step property of the NumericUpDownExtender control. 184

#### **Chapter 15: Deployment**

1. What is deployment? 185
2. List different ways of deployment that are supported by .NET Framework 4.0. 185
3. What is XCOPY? 185
4. Does XCOPY copy the hidden and system files? 185
5. Why do you use Windows Installer? 186
6. What is ClickOnce? 186
7. What are the enhancements in ClickOnce deployment in .NET 4.0? 187
8. What is the difference between deploying and publishing an application? 187
9. How can you determine whether you should deploy the application or publish the application? 188
10. Explain the .NET Framework deployment features. 188
11. What do you mean by Merge Module projects? 188
12. What is the need of Copy Web Site? 188

13. How can you deploy an ASP.NET Web application? 189
  14. Can you deploy an ASP.NET Web application project using the Copy Web Site option? 189
  15. Can Windows applications and the Web applications be deployed using the same template of Setup and Deployment project? 189
  16. What is the use of the Copy Project command? 189
  17. What is Application Cache? 190
- Chapter 16: .NET Assemblies**
1. What is an assembly? 191
  2. Name the different components of an assembly. 191
  3. What are the different types of assemblies? Explain them in detail. 191
  4. Can one DLL file contain the compiled code of more than one .NET language? 192
  5. What is the maximum number of classes that can be contained in a DLL file? 192
  6. What is Assembly Manifest? 192
  7. What is metadata? 193
  8. The AssemblyInfo.cs file stores the assembly configuration information and other information, such as the assembly name, version, company name, and trademark information. (True/False). 193
  9. What is GAC? 194
  10. What is the value of the Copy Local property when you add an assembly in the GAC? 194
  11. Is versioning applicable to private assemblies? 194
  12. Discuss the concept of strong names. 194
  13. What is the significance of the Strong Name tool? 194
  14. Where is the information regarding the version of the assembly stored? 195
  15. Name the MSIL Disassembler utility that parses any .NET Framework assembly and shows the information in human readable format. 195
  16. What is Native Image Generator? 195
  17. Which utility allows you to reference an assembly in an application? 195
  18. How do you add and remove an assembly from GAC? 195
  19. What is a satellite assembly? 195

20. How can different versions of private assemblies be used in the same application without a re-build? 196
21. What is the difference between EXE and DLL? 196
- Chapter 17: Remoting and Reflection**
1. Explain .NET remoting. 197
  2. Define activation. 197
  3. What is a Windows Process? 197
  4. How .NET remoting is different from Distributed Component Object Model (DCOM)? 197
  5. What do you mean by an application domain? 197
  6. List the activation modes in which you can create a remote object. 198
  7. Explain the .NET remoting architecture. 198
  8. What is the difference between remotable and nonremotable objects? 199
  9. Define marshalling. 200
  10. Differentiate between the marshal-by-value and marshal-by-reference objects. 200
  11. Do marshal-by-value objects require an activation process? 200
  12. What are well-known objects? 200
  13. What are context-bound objects? 201
  14. What do you mean by server-activated objects? 201
  15. What are the two modes to activate the SAOs? 201
  16. What is the difference between the Singleton mode and SingleCall mode? 201
  17. Describe client-activated objects (CAOs). 201
  18. What is a channel? List different types of channel. 202
  19. Explain the TCP channel. 202
  20. What is the difference between HTTP and IPC channels? 202
  21. What are message sinks? 203
  22. What are different types of message sinks? 203
  23. What are formatters? List different types of formatters. 204
  24. Give a brief description about BinaryFormatter and SOAPFormatters. 204
  25. What do you mean by proxies? 204
  26. Define transparent and real proxies. 205
  27. How can you distinguish between Web services and remoting? 205

28. How can you decide to use the singleton mode in remoting? 206
29. Are client-activated objects stateful in nature? 206
30. Which configuration file contains all the supported channels? 206
31. Is it appreciated to distribute the implementation to remoting clients? If no, why? 206
32. How is a method called asynchronously in remoting? 206
33. Which class do we inherit while making an object remotable? 206
34. What are the different ways to host a remote object? 207
35. What do you mean by the lifetime (lease time) of a remote object? 207
36. How is the lease of an object defined? 207
37. Is it possible to configure a .NET Remoting object by using an XML file? 207
38. Which Microsoft tool is used to automatically generate interface for the remotable object in .NET? 207
39. What is WSDL? 207
40. What is the difference between the RegisterWellKnownServiceType() and RegisterWellKnownClientType() methods in .NET remoting? 207
41. Do any security measures exist for .NET Remoting in the System.Runtime.Remoting namespace? 208
42. What is reflection? 208
43. Why do we need reflection? 208
44. How are application, process, and application domains are related? 208
45. When will you use .NET Remoting and when will you use ASP.NET Web Services? 209
46. Which classes are used by reflection to get information about various types? 209
- Chapter 18: ASP.NET MVC Framework**
1. What is Model-View-Controller (MVC)? 211
  2. What is ASP.NET MVC Framework? 211
  3. What are the features of ASP.NET MVC Framework? 211
  4. Explain the components of ASP.NET MVC Framework. 212
  5. Explain the interaction between the three components of the MVC architecture. 213
  6. Which is the base class of ASP.NET MVC Framework? 214

7. Which class defines the preconfigured URL routes for the ASP.NET MVC Framework? 214

8. What is URL Routing in ASP.NET MVC? 214

9. In the MVC architecture, data (Model) and user interface (View) are separated from each other so that any change in the user interfaces does not affect the data in the database. (True/False) 214

10. What is TDD environment? 214

11. ASP.NET MVC Framework provides an alternative to the developer while designing and developing a Web as well as Windows applications. (True/False). 215

12. Which is the latest version of ASP.NET MVC that is built-in in Visual Studio 2010? 215

13. What are the changes introduced in the ASP.NET MVC 2 framework? 215

14. Explain the new features included in MVC 2 Framework. 215

15. Explain the T4 template included in the ASP.NET MVC 2 Framework. 216

16. What are new improvements in ASP.NET MVC 2 API? 216

#### *Chapter 19: Cloud Computing*

1. What is cloud computing? 217

2. What is a cloud? 217

3. What are the basic characteristics of cloud computing? 217

4. Explain the cloud architecture. 218

5. What is a Cloud Service? 219

6. What are main features of cloud services? 219

7. What are the advantages of cloud services? 220

8. Explain different models of cloud service. 220

9. How many types of deployment models are used in cloud? 221

10. Explain public and private cloud. 221

11. Explain hybrid and community cloud. 222

12. Why does an organization need to manage the workloads? 222

13. Give a brief introduction of Windows Azure operating system. 222

14. Which services are provided by Window Azure operating system? 223

15. What is the AppFabric component? 224

## Glossary

### .NET Framework Class Library

The .NET Framework Class Library (FCL) is a library of reusable types such as classes, structures, interfaces, and enumerations that can be used in the .NET code.

### Abstract class

An abstract class is a class that cannot be instantiated, meaning that you cannot create objects of the class.

### Activity

Activities are the basic building blocks of workflows built on Windows Workflow Foundation. They are standalone pieces of information that can be reused multiple times within a workflow or across multiple workflows.

### ADO.NET Entity Framework

The ADO.NET Entity Framework allows you to focus on data through an object model instead of a relational model. It requires you to write less data access code, reduce maintenance, and facilitates the persistence of data.

### AJAX

AJAX is a group of interrelated web development techniques used for creating interactive Web applications or rich Internet applications.

### AJAX Control Extender Toolkit

AJAX Control Toolkit is a set of extenders that are used to extend the functionalities of the ASP.NET controls.

### Anonymous types

An anonymous type provides you a convenient way to summarize a set of read-only properties into a single object without defining a type.

### Application Domain

An application domain is an entity inside which every .NET application runs. Each application domain executes a single application. An operating system process hosts an instance of the CLR and each instance of CLR can manage multiple application domains.

### Application Domain Host

An application domain host is responsible for loading the CLR into a process, creating application domains within the process, and loading the application code into the application domains.

### Application Package

An application package is a file with .xap extension that consists of all the necessary files and resources to run a Silverlight application.

## **Assembly**

Assemblies are packaged into units containing programs and libraries. An assembly contains a self-describing binary file, which can be either Dynamic Link Library (DLL) or Executable (EXE).

## **Assembly manifest**

Assemblies maintain all their information in a special unit called the assembly manifest.

## **Asynchronous Postback**

Asynchronous postback executes only one postback at a time that means only the required part of the page is updated.

## **Binding**

A binding is an object that indicates how the clients connect and communicate with the service endpoint. Bindings can include the transport protocol, encoding scheme, and security mechanism required to communicate with the service endpoint.

## **Bitmap Image**

Bitmap image is a type of memory organization or image file format that is used for storing the digital images. This term originates from terminology of computer programming that implies a map of bits or mapped array of bits.

## **ClickOnce**

ClickOnce is a deployment technique, which allows you to deploy self-updated Windows and Console applications that can be installed, updated, and run from a Web site.

## **Code access security**

Code access security (CAS) is part of the .NET security model that determines whether or not a piece of code is allowed to run and what resources it can use while running.

## **Common Language Runtime (CLR)**

The Common Language Runtime (CLR) hosts a variety of languages and ensures interoperability among code from different language compilers.

## **Common Language Specification**

Common Language Specification (CLS) is a set of basic rules, which enables interoperability between two .NET-compliant languages.

## **Common Type System (CTS)**

Common Type System (CTS) specifies the guidelines for declaring, using, and managing types at runtime.

## **Container Control**

A container control is a WPF control that contains other controls (or element) called child controls (or elements). The Grid, StackPanel, and Canvas are some commonly used container controls.

## **Control Template**

A control template is a WPF object that defines the appearance, style, and behavior of a control. Most WPF controls have a default control template; however, new control templates can be created to give a new look and behavior to a control.

## **Copy Project**

Copy Project is a command that copies only the files required to run the project and pastes it on the target server.

## **Copy Web Site**

The Copy Web Site tool is provided with Visual Studio 2008, which enables you to copy a Web site from your development server to either the same server or a remote server.

## **Data Contract**

A data contract is an agreement between a client and a service that conceptually depicts the data to be exchanged. Data contracts define the data types that are used in the service.

## **Data Template**

A data template is an object that provides a means to change the presentation of the data or content of a WPF control.

## **Dataset**

A dataset stores data in a disconnected cache. The structure of a dataset is similar to a relational database. A dataset gives you access to an object model of tables, rows, and columns and contains constraints and relationships defined for the dataset.

## **Dynamic Resource**

A dynamic resource is a WPF resource that uses the DynamicResource markup extension. With dynamic resources as property values, you can defer the assignment of a property value until run time.

## **Delegate**

A delegate is a special type of object that contains the details of a method rather than data.

## **Endpoint**

An endpoint is a connecting point between a WCF service and its clients. A WCF service exposes its

## **Dependency Property**

A dependency property is a property that depends on various factors or inputs for its value. The other factors or inputs may include data-bound values, values set through styles, animations, and resources, and system-specific or user-specific factors. This implies that the value of a dependency property can be set directly or indirectly through other inputs.

## **Deployment**

Deployment refers to the distribution of an application among various end-users.

## **DOM**

Document Object Model (DOM) defines the logical structuring of XML documents and the way a document is accessed and manipulated.

## **DTD**

A Document Type Definition (DTD) defines the structure of the content of an XML document, thereby storing the data in a consistent format.

**functionalities to the clients through its endpoints.** A single WCF service can have one or more endpoints. Each endpoint is characterized by an address, binding, service contract, and behavior.

### Event

An event is a delegate type class member that is used by an object or a class to provide a notification to other objects that an event has occurred.

### Event Handler

An event handler is a method that contains the code to handle a particular event. In most cases, an event handler is a Sub procedure. There can be multiple event handlers for a single event.

### Exception

An exception is a runtime error that arises because of some abnormal conditions.

### Extension methods

An extension method enables you to add methods to existing types without creating a new derived type or modifying the original type.

### Fixed Document

A fixed document is a WPF document in which the arrangement and positioning of the entire content is static or fixed. In other words, the layout of the entire content of fixed documents remains the same irrespective of the screen or print device used.

### Flow Document

A flow document is a WPF document in which the content can be re-arranged such that optimum readability is achieved. The rearrangement of the content is based on various factors, such as the window size, display resolution, and print resolution. Fixed documents have a flexible and dynamic layout.

### Global Assembly Cache (GAC)

Global Assembly Cache (GAC) is an assembly reservoir that stores all the assemblies, which are supposed to be shared by a number of applications on a machine.

### Global.asax File

The Global.asax file is a component of ASP.NET applications. This file contains code for the events that are raised at the execution time of the application.

### Host Process

Host process is the process within which a workflow is hosted and run. A host process may be a Windows Forms application, a Web application, or a Web service application.

### Human Workflow

Workflows that are intended to coordinate interactions among people are known as human workflows.

### Hyperlink

Hyperlink is a reference or a link that is present on the document to refer to another document or different section of the same document.

### IIS

It is an acronym for Internet Information Services, used to host websites. It is also called a Web server.

### Impersonation

Impersonation is a technique to execute a client's request by using the identity of the client who is making the request.

### Inheritance

- Inheritance is the property through which a child class obtains all the features defined in its parent class. A parent class is one which is at a higher level in the class hierarchy than the child class.

### Interface

Interfaces are the means of implementing the concept of multiple inheritance. You cannot have a class that inherits from more than one base class; however, you can implement more than one interface in a single class.

### Markup Extension

Managed code is the code that is designed to run under the CLR. Any code written in Visual Basic 2008 is managed code, by default.

### Lambda expressions

A lambda expression is an anonymous method that contains either an expression or a block of statements.

### LINQ

Language-Integrated Query (LINQ) is a new component of .NET Framework 3.5 that offers a unique way of querying data from diverse types of data sources.

### Managed Code

Managed code is the code that is designed to run under the CLR. Any code written in Visual Basic 2008 is managed code, by default.

### Metadata

A markup extension is a means of setting the value of a WPF property to an object reference rather than to a string or numeric value. It also allows you to delay the assignment of a value to a property until the runtime or to bind a property to some data.

### Marshalling

It is a process for packaging and sending method calls between the objects and across application boundaries by using serialization and deserialization.

### Just-In-Time Compiler

Just-In-Time (JIT) compiler is the compiler that converts MSIL code into machine code at the point when the code is required at run time.

.NET assemblies have some metadata associated with it that includes information about each type (class, structure, and enumeration), defined in the assembly, and the members (properties, methods, and events) of the types.

<b>Microsoft Intermediate Language (MSIL)</b>	and the processing boundaries of an application.
<b>(MSIL)</b>	is used as the output of a number of compilers and as the input to the Just-In-Time (JIT) compiler.
<b>MVC</b>	Model-View-Controller (MVC) is a design pattern used in software engineering which separates data access, business logic, and presentation logic.
<b>Native Image Generator</b>	The Native Image Generator (Ngen.exe) is a tool used by managed applications to improve the performance.
<b>Private Assembly</b>	A private assembly is a collection of modules that is used only by the application with which it has been deployed.
<b>Reference Type</b>	A reference types refer to the data type where a variable of that type stores the address to another variable that contains the value.
<b>Reflection</b>	Reflection is a process of finding out the internals (such as metadata, assemblies, modules, and types) of an application without accessing the source code.
<b>Remoting</b>	Remoting provides a powerful programming model and runtime support to make objects available outside the machine boundaries

<b>Routed event</b>	A routed event refers to those events that route or traverse through the element tree of a WPF application.
<b>RTF file</b>	RTF files are the text files with advanced formatting options. These files are provided with an extension .rtf.
<b>Runtime Engine</b>	A runtime engine runs within the host process and is responsible for executing each workflow instance.
<b>ScriptManager</b>	The ScriptManager control renders the AJAX Library scripts to the browser and supports for partial page rendering, Web service calls, and use of ASP.NET AJAX Client Library.
<b>Sequential Workflow</b>	A sequential workflow is intended for applications where activities of the workflow are executed in a well-defined order.
<b>Service Contract</b>	A service contract defines the entire functionality of a WCF service that is exposed to the clients. A service contract groups the various operations and defines the message exchange pattern and the message format.
<b>Smart Device Application</b>	Applications developed to run on mobile devices, such as Personal Digital Assistants (PDAs) and Smartphones, are referred to as smart device applications. Such applications can be developed on

<b>Shared assembly</b>	A shared assembly is a collection of types and resources contained within some modules that is used by several clients on a single machine.
<b>Side-by-side execution</b>	Side-by-side execution is the ability to run multiple versions of an application or component on the same computer.
<b>SingleCall</b>	It is a mode that instantiates an object to service a single client request.
<b>Singleton</b>	It is a mode in which one object services the requests of all clients by maintaining state across requests and method calls.
<b>Smart Device</b>	A smart device is a small, portable device that is capable of running programs and connecting to a computer network. A smart device has a CPU for executing programs, a random Access Memory (RAM) for storing program and data during execution, and a persistent storage, such as hard disk for permanently storing program and data.

<b>SOAP</b>	Simple Object Access Protocol (SOAP) is a standard communication protocol which is used for transferring data among applications that execute on diverse platforms or hardware.
<b>State Machine Workflow</b>	A state machine workflow models its activities into a finite state machine. It contains a set of states, each of which can receive a specific set of events.
<b>Static Resource</b>	A static resource is a WPF resource that is referenced by using the StaticResource markup extension. The value of a static resource is determined at the time of loading.
<b>Strong Name</b>	A name that comprises of an assembly's identity—it's text name, version number, and culture information strengthened by a public key and a digital signature.
<b>System Workflow</b>	A system workflow is a workflow that coordinates interactions among applications.
<b>Test page</b>	A test page is the HTML page that hosts the Silverlight plug-in in an <object> tag. The test page is created when the Silverlight application is compiled and is displayed to the users when the application is accessed.

**Native Image Generator**

The Native Image Generator (Ngen.exe) is a tool used by managed applications to improve the performance.

**Trigger**

A trigger is an object that causes an action when a particular condition is met. Triggers can be property triggers (those triggers that set a property value when there is a change in some other property value) or event triggers (those triggers that perform an action when an event occurs).

**UDDI**

Universal Description, Discovery, and Integration (UDDI) is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet.

**Unmanaged Code**

Unlike managed code, unmanaged code is not verifiable by CLR and performs tasks, such as pointer arithmetic, and thus manages memory on its own.

**UpdatePanel**

The UpdatePanel control defines portions or regions of a page that can be updated together when a postback is triggered.

**Value Type**

A value type refers to the data type where a variable of that data type stores its value in the memory location assigned to the variable.

**Web Application**

It is an application that is hosted on the Internet and can be accessed through a Web browser and used by multiple users simultaneously.

**Web Server Controls**

The ASP.NET Web server controls are objects on the ASP.NET pages that run when the Web page is requested and that render markup to the browser.

**Web Service**

A Web service may be defined as an independent and self-sustained unit of a software application that is hosted on the Web and implements specific functionalities to execute the business logic.

**web.config**

The web.config file applies configuration settings to the directory it is located in and to all virtual child directories beneath it. The settings in child directories can optionally override or modify the settings specified in parent directories.

**Windows Installer**

Windows Installer is a software installation and configuration service that ships with operating systems. The Windows Installer Service contains information about every application it installs.

**Windows services**

Windows services are applications that are installed as system services and these services continuously run in the

**background.**

These services automatically start when the computer boots and do not have any user interface.

**Workflow**

A workflow is a graphical representation of the flow of a process or a set of tasks that produces a result.

**WSDL**

Web Services Description Language (WSDL) is used to describe a Web service in terms of the messages that it creates and accepts. The WSDL document is an XML file that contains the interface schema for the Web service.

**XAML**

XAML or Extensible Application Markup Language is a declarative markup language based on XML. XAML is supported by many technologies, such as WPF, WF, and Silverlight. In XAML, various predefined markup tags (or elements) and their attributes that can be used to declaratively design the user interface of applications.

**XAP**

An XBAP or XAML browser application is a WPF application that is hosted on a Web server and is accessed through a Web browser, such as Internet Explorer and Firefox.

**XCopy**

XCopy is a command used to deploy an application by copying a set of files to a folder on the target

**computer.**

and then executing the application at the client side.

**XML**

Extensible Markup Language (XML) is a markup language that enables to store and exchange data in a structured format across different platforms and devices.

**XML namespace**

An XML namespace is used to avoid dispute in locating XML elements of same name that are contained in more than one XML document.

**XML Schema**

An XML schema defines the structure of an XML document and also defines the order and list of elements and attributes that can be used in an XML document.

**XmIHttpRequest**

XmIHttpRequest is used by XMLHttpRequest to transfer XML and other text data between client and server.

**XPath**

XML Path (XPath) is a language used to access different parts of an XML document, such as elements and attributes.

**Cloud**

A cloud is a combination of hardware, networks, storage, services, and interfaces that helps in delivering computing as a service.

**IaaS**

This model of cloud service provides storage and compute resources that developers and IT organizations can use to offer customized business solutions.

**SaaS**

This model of cloud service delivers the software as a service over the Internet.

**AppFabric**

The AppFabric component is used to create access control and distribute messages across clouds and enterprises. It has a service-oriented architecture, and can be considered as the backbone of the Windows Azure platform.

**PaaS**

This model of cloud service enables the provider to deliver the additional amount of infrastructure.

**XSLT**

Extensible Stylesheet Language Transformations (XSLT) is an XML-based language that enables the transformation of an XML document into another XML or HTML document.