KKday .Net Interview Questions

1. **What is Difference between NameSpace and Assembly?**

Assembly is physical grouping of logical units, Namespace, logically groups classes.  
Namespace can span multiple assembly.

1. **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.

1. **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.

1. **What is the function of the ViewData property?**

ViewData is similar to ViewBag. It is useful in transferring data from Controller to View.

ViewData is a dictionary which can contain key-value pairs where each key must be string.

1. **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.

1. **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.

1. **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.

1. **What is serialization?**

**Serialization is the process of converting an object into a stream of bytes.Deserialization is the opposite process, i.e. creating an object from a stream of bytes. Serialization/Deserialization is mostly used to transport objects (e.g. during remoting), or to persist objects (e.g. to a file or database).**

1. **What is the managed and unmanaged code in .net?**

The .NET Framework provides a run-time environment called the Common Language Runtime, which manages the execution of code and provides services that make the development process easier. Compilers and tools expose the runtime's functionality and enable you to write code that benefits from this managed execution environment. Code that you develop with a language compiler that targets the runtime is called managed code; it benefits from features such as cross-language integration, cross-language exception handling, enhanced security, versioning and deployment support, a simplified model for component interaction, and debugging and profiling services.

1. **How do you create a permanent cookie?**

Setting the Expires property to MinValue means that the Cookie never expires.

1. **Difference between Class And Interface in .NET?**

Class is logical representation of object. It is collection of data and related sub procedures with definition.

Interface is also a class containing methods which is not having any definitions.

Class does not support multiple inheritance. But interface can support

1. **Can any object be stored in a ViewData in .NET?**

An object that either is serializable or has a TypeConverter defined for it can be persisted in ViewData.

1. Explain the difference between a class and an object.

In short, a class is the definition of an object, and an object is instance of a class.

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

1. Explain the difference between boxing and unboxing. Provide an example.

Boxing is the process of converting a value type to the type object, and unboxing is extracting the value type from the object. While the boxing is implicit, unboxing is explicit.

Example (written in C#):

int i = 13;

object myObject = i; // boxing

i = (int)myObject; // unboxing

1. What do you understand by LINQ? Enlist its types.

Language Integrated Query is the collection of standard query operators which adds query facilities into .NET framework languages like C#, VB.NET, etc. LINQ also represents the set of method names with the translation rules that can be used by the compiler for converting fluent-style query expressions into expressions that uses these method names, anonymous types and lambda expressions. LINQ can be used to extract data from arrays, XML documents, relational databases and other third-party data sources.

The LINQ is of the following types:

LINQ to objects refers to the use of LINQ queries without the use of an intermediate LINQ provider such as LINQ to SQL.

* LINQ to XML which was formerly known as XLINQ
* LINQ to SQL which was formerly known as DLINQ
* LINQ to Datasets
* LINQ to Entities

1. Explain the differences between an Interface and an Abstract Class in .NET.

An interface merely declares a contract or a behavior that implementing classes should have. It may declare only properties, methods, and events with no access modifiers. All the declared members must be implemented.

An abstract class provides a partial implementation for a functionality and some abstract/virtual members that must be implemented by the inheriting entities. It can declare fields too.

Neither interfaces nor abstract classes can be instantiated.

1. What is a delegate in .NET?

A delegate in .NET is similar to a function pointer in C or C++. Using a delegate allows the programmer to encapsulate a reference to a method inside a delegate object. The delegate object can then be passed to code which can call the referenced method, without having to know at compile time which method will be invoked. In addition, we could use delegate to create custom event within a class. For example,

public delegate void FooDelegate();

class FooClass

{

// custom event

public event FooDelegate FooEvent;

}

FooClass FooObj = new FooClass()

FooObj.FooEvent += new FooDelegate();

1. Difference between FirstOrDefault and First in LINQ

When you use the First method, you expect to find a result. If there is no result, the First method will raise an exception. When you use the FirstOrDefault method, if no value so found, null is returned instead of an exception being raised. You will therefore use the First method when your business rules say that a result must exist and it would be an exception if one doesn't exist. You would use FirstOrDefault if you are not certain of returning a result, and no result is acceptable according to your business rules.

1. What is ASP.NET Core?

ASP.NET Core 1.0 is the next version of ASP.NET. It is open source and cross-platform framework (supports for Windows, Mac and Linux) suitable for building cloud based internet connected applications like web apps, IoT apps and mobile apps. ASP.NET Core apps can run on .NET Core or on the full .NET Framework.

1. What is Startup.cs file in ASP.NET Core?

In ASP.NET, Global.asax (though optional) acts as the entry point for your application. Startup.cs, it is entry point for application itself. The Startup class configures the request pipeline that handles all requests made to the application. Read this read this excellent post The Startup.cs File in ASP.NET Core 1.0 – What Does It Do? to know more about startup.cs.