**1.What is .NET?**

NET is an integral part of many applications running on Windows and provides common functionality for those applications to run. This download is for people who need .NET to run an application on their computer. For developers, the .NET Framework provides a comprehensive and consistent programming model for building applications that have visually stunning user experiences and seamless and secure communication.

**2.How many languages .NET is supporting now?**

When .NET was introduced it came with several languages.

[VB.NET](http://vb.net/),

C#,

COBOL

and

Perl, etc.

**What is the difference between Response.Redirect and Server.Transfer?**

Response.Redirect basically redirects the user’s browser to another page or site. The history of the user’s browser is updated to reflect the new address as well. It also performs a trip back to the client where the client’s browser is redirected to the new page.

Whereas, Server.Transfer transfers from one page to the other without making any round-trip back to the client’s browser. The history does not get updated in the case of Server.Transfer.

|  |  |
| --- | --- |
| **Class** | **Object** |
| Class is the definition of an object | An object is an instance of a class. |
| It is a template of the object | A class does not become an object unless instantiated |
| It describes all the methods, properties, etc | An object is used to access all those properties from the class. |

**what do you know about boxing and unboxing?**

|  |  |
| --- | --- |
| **Boxing** | **Unboxing** |
| Implicit | Explicit |
| Converting a value type to the type object | Extracting the value type from the object |
| eg : obj myObject = i; | eg : i = (int)myObject; |

**Differentiate between constants and read-only variables.**

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

**From which base class all web Forms are inherited?**

All web forms are inherited from **page class**.

**What are the different types of constructors in c#?**

Following are the types of constructors in C#:

* Default Constructor
* Parameterized constructor
* Copy Constructor
* **public** **class** Employee
* {
* **public** string firstName;
* **public** string lastName;
* **public** string position;
* **public** int salary;
* **public** Employee()
* {
* }
* // Copy constructor.
* **public** Employee(Employee employee)
* {
* firstName = employee.firstName;
* lastName  = employee.lastName;
* position  = employee.position;
* salary    = employee.salary;
* }
* }
* Static Constructor

**Characteristic of static constructor**

1. A static constructor does not take any access modifiers.
2. A static constructor does not have a parameter.
3. A static constructor is called automatically to initialize the class before the first instance is created or any static members are referenced.
4. A static constructor cannot be called directly.
5. The user has no control over when the static constructor is executed in the program.
6. A typical use of static constructors is when the class is using a log file and the constructor is used to write entries to this file.
7. A class can have only one static constructor.
8. It can access only static members of a class.

* Private Constructor

A private constructor is a special instance constructor. It is generally used in classes that contain static members only. If a class has one or more private constructors and no public constructors, other classes (except nested classes) cannot create instances of this class. The use of private constructor is to serve singleton classes. A singleton class is one which limits the number of objects created to one. Using private constructor we can ensure that no more than one object can be created at a time

* One use of private constructor is when we have the only static member.
* It provides the implementation of singleton class pattern.
* Once we provide constructor (private/public/any) the compiler will not add the no parameter public constructor to any class.

**use of static constructor. Why and when would we create a static constructor and is it possible to overload one?**

No you can't overload it; a static constructor is useful for initializing any static fields associated with a type (or any other per-type operations) - useful in particular for reading required configuration data into readonly fields, etc.

It is run automatically by the runtime the first time it is needed (the exact rules there are complicated (see "beforefieldinit") and changed subtly between CLR2 and CLR4). Unless you abuse reflection, it is guaranteed to run *at most* once (even if two threads arrive at the same time).

**List the events in the page life cycle.**

Following are the events in the page life cycle:

* Page\_PreInit
* Page\_Init
* Page\_InitComplete
* Page\_PreLoad
* Page\_Load
* Page\_LoadComplete
* Page\_PreRender
* Render
* **What is the code to send an email from an ASP.NET application?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | mail message = new mail();  message.From = "abc@gmail.com";  message.To = "xyz@gmail.com";  message.Subject = "Test";  message.Body = "hello";    SmtpMail.SmtpServer = "localhost";  SmtpMail.Send(message); |

**What is cross-page posting?**

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

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

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

**What are the different types of cookies in ASP.NET?**

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

**What is the difference between ExecuteScalar and ExecuteNonQuery?**

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

**What is the difference between a stack and a heap?**

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