**DOTNET(.NET)**

***INTRODUCTION***

* Is not a programming language
* It is a frame work.
* Where we can develop development, deployment, console, web, windows applications & services.

**C#**

* It is an object-oriented programming language.
* Easy to learn & simple to use.
* **Console application** :- It is a character user interface application.
* **Windows application** :- Graphical user interface.
* **Web applications** :**-** Graphical user interface & shared application.
* **MSIL**:-Micro Soft Intermediate Language code is partially compiled code.
* **CLS**:-Set of rules defined by Microsoft to allow language independent in dotnet.
* **CTS**:-Common Type System ensures that Data types defined in 2 Different languages gets compiled to a common datatype.
* **JIT**:-Just In Time compile IL code to machine language.
* **CLR**:-Common Language Runtime invokes JIT to compile IL code.

**What is name space in c#**

* C# name space are used to logically arrange classes structs, interfaces, Enums[An enum is a special "class" that represents a group of **constants]** and delegates.
* Name space is used to declare a scope that contains a set of related objects.

**CLASS**

* Template R blue print for creating an objects.
* It contains variables, methods, functions, properties, constructors, delegates & indexers.
* Just having a class does not allocate any memory.
* We have written everything in the class only.
* Definitely the class should be under namespace

**OBJECT**

* Instance of that particular class.
* Created by using the “new” keyword.

**VARIABLES**

To store some values we use variables.

**METHODS**

• Methods can be defined within a class.

• Methods can have parameters & return the values.

• Methods can also declare exceptions.

**FUNCTIONS :**

• Function is a named block of code that performs specific operation & return a value.

• improve code reusability

**PROPERTIES :**

• We use properties to write our own(Or) Logic implementation.

• We use “get “&”Set” as properties.

**TYPE CASTING**

* Converting one datatype into another datatype. 2types of casting
* **Implicit casting** :-converting a smaller size type into larger size type.
* **Explicit casting** :-converting a larger size type into smaller size type.

**KEYWORDS**

* **This keyword** : To refer the particular current class object
* **Base keyword** : To access the base class members into the derived class.
* **New keyword** : Hiding base class members in derived class.
* **Virtual keyword**: Virtual keyword helps us to define some logic in the parent class which can be overridden in the child class.

**INCREMENTS**

* **Pre increment** : same line gets increment.
* **Post increment** : next line gets increment.

**LOOPINGS**

* **FOR**
* **FOREACH**
* **WHILE**
* **DO WHILE**

FOR LOOP : To iterate a part of the program in several times.

For(Initialization; condition; increment R decrement)

{

Statement

}

FOREACH : For each iterates items in a collection like an array list.

Foreach(var name in object )

{  
statement

}

WHILE LOOP : It’s check the condition first then the loop body is executed.

While <condition>

{

Statement

}

DO WHILE LOOP : It’s executes the body then checks the condition

Do

{  
statements

}

while condition;

**DIFFRENCES BETWEEN ARRAY & ARRAY LIST**

|  |  |
| --- | --- |
| ARRAY | ARRAY LIST |
| 1.It is a collection of similar data type items | **1.It is not a collection of similar data type items** |
| 2.It is static | **2.It is dynamic** |
| 3. It is fixed sized data structure (strongly type) | **3. Not a fixed size data structure (not strongly type)** |

**CONTROL STATEMENT**

* **IF**
* **IF ELSE**
* **ELSE IF,**
* **SWITCH**

**IF** : If the condition is true the statement will be executed. If it is false the statement will not

Execute.

If (condition)

{

**Statement**

}

**IF ELSE** : it is used to checking whether a condition is true R false

If (condition)

{

statement1

}

**Else**

{

statement2

}

**SWITCH :**  It is used to select one of many statement blocks to be execute

Switch(expression)

{

Case1:  
statement

Break;

Case2:

Statement

Break;

Default:

Break;

}

Break:-Terminate the loop conditions.

**DIFFERENCES BETWEEN STACK &HEAP**

|  |  |
| --- | --- |
| STACK | HEAP |
| 1.Stack stores datatypes like int ,double ,bool etc. | **1.Heap stores datatypes like string & obj.** |

**DIFFERENCES BETWEEN VAR & DYNAMIC**

|  |  |
| --- | --- |
| VAR | DYNAMIC |
| 1.The type variable is decided by the compiler at compile time. | **1.The type of the variable is decided at the run time** |
| 2.var we can convert that particular datatype. | **2.dynamic we can’t be converted that particular datatype.** |

**DIFFERENCESBETWEEN STRING vs STRING BUILDER**

|  |  |
| --- | --- |
| STRING | STRING BUILDER |
| 1.String is immutable( Do not change) | **1String Builder is mutable. (we can change )** |
| 2. new object will be created | **2. new object will not be created** |

**DIFFERENCESS BETWEEN VALUETYPE AND REFENCE TYPE**

|  |  |
| --- | --- |
| VALUE TYPE | REFERENCE TYPE |
| 1.It is actual datatype. | **1 It is a pointer data.** |
| 2.To stored on stock | **2.stored on heap** |
| 3.Normal datatypes like int bool. | **3.Are all objects.** |

**DEBUGGING**

* Locating and correcting code Errors in a computer programme.
* Debugging allows the developers to see how the code works in a step by step manner how the objects  are created and destroyed etc.

**ACCESS MODIFIERS**

we use access modifiers to secure the program.

* **PUBLIC : we can access public members at anywhere.**
* **PRIVATE : we can access private members within the class only not outside of the class.**
* **INTERNAL : we can access internal members within the assembly (particular project).**
* **PROTECTED : it is just like a private , derived class can access protected members. but**

**we cant access protected members out side of the class.**

**Differences between Console Read & Console Read Line**

|  |  |
| --- | --- |
| Console Read | Console Read Line |
| Console Read is used to read only a single character from the standard output device. | **Console Read Line is used to read a line or string from the standard output device.** |

**Garbage Collection**

* Garbage collection is a background process which cleans unused managed resources.
* The Garbage collection manages the allocation and releases of memory for application.

**Differences between managed code & un managed code**

|  |  |
| --- | --- |
| Managed Code | Un Managed code |
| 1.Code that executes under CLR is called managed code. | **1.Code that executes outside CLR boundary is called un managed code** |
| 2.The managed code provides security to the code. | **2.The un managed code creates security threads.** |
| Ex :-C#,F#,VB | **Ex:-C++,VB6,V++** |

**CLASS LIBRARIES**

• It is a collection of reusable , object oriented class libraries including classes, interfaces, data types & methods.

• .net standard set of libraries , referred as base class libraries (BCL-Core set) (or) framework class libraries (FCL-complete set)

• Framework class library is nothing but a collection of namespace & classes, which can be reusable.

• These framework class libraries are built in…

• Example : system, system. Data, etc

**OOPS CONCEPTS**

OOPS – Object Oriented Programming language. Is based on the concepts and objects.

OOPS have four important pillars

* **ENCAPSULATION**
* **ABSTRACTION**
* **INHERITANCE**
* **POLYMORPHISM**

**ENCAPSULATION**

Encapsulation is defined as the wrapping up of the data under a single unit. Wrapping means combine.

**ABSTRACTION**

Abstraction in C# is the process of hide the internal details and show only the functionality.

**INHERITANCE**

Acquires base class members into the derived class members.

There are two types of Inheritance

**1.MULTIPLE INHERITANCE**

**2.MULTILEVEL INHERITANCE**

1**.MULTIPLE INHERITANCE** :-Multiple inheritance is only possible by Interfaces not by using in classes.

**2.MULTILEVEL INHERITANCE**:-Multilevel Inheritance is possible in classes

**POLYMORPHISM**

* Which we can perform a single action by different ways.
* Polymorphism is derived from two Greek words. POLY and MORPHS.
* The word” poly” means many ”morph” means forms.
* So polymorphism means many forms.

**Static polymorphism:-** It is implemented by method overloading.

**Dynamic polymorphism:-**It is implemented by method overriding.

**Method Overloading:-**Same method name with different signature in the class.

**Operator Overloading:-** Over loading helps to redefine additional functionalities for plus, minus,

multiplication.

**Overriding**:-Changing the implementation of base class member to the derived class member is called overriding

**Differences between Interface and abstract base class**

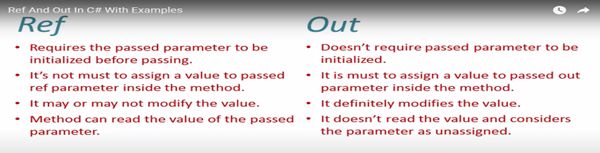
|  |  |
| --- | --- |
| INTERFACE | ABSTRACT BASE CLASS |
| 1.Interface with contain with out implementation. | **1.Abstract class contain both with implementation & without implementation.** |
| 2.An Interface can inherit multiple interfaces but cannot inherit a class. | **2.An abstract class can inherit one class only and it can inherit multiple interfaces.** |
| 3.An interface cannot declare constructors. | **3.An abstract class can declare constructors.** |
| 4.An interface method is can implemented without using the override keyword. | **4.An object is implemented by using the override keyword.** |

**Differences between Constant & Read only**

|  |  |
| --- | --- |
| CONSTANT | READ ONLY |
| 1.Constant is a static for program | **1.Read only is a static for an object.** |
| 2.Constant variable must be assigned a value at the time of declaration | **2.Readonly variables is a runtime constant .Its value can be set only in the constructor or as part of an object initializer** |
| 3.The value of a constant variable is known at compile time and cannot be changed at runtime. | **3.The value of a read only variable is known only at runtime.** |

**DIFFERENCES BETWEEN REF AND OUT**

When we want to return more than one value from method then we can use the out and ref parameters.



**CONSTRUCTORS**

To initialize the class members we can use constructors.

There are Five types

* **DEFAULT CONSTRUCTOR**
* **PARAMETARISED CONSTRUCTOR**
* **STATIC CONSTRUCTOR**
* **PRIVATE CONSTRUCTOR**
* **COPY CONSTRUCTOR**

1.DEFAULT CONSTRUCTOR:-A Constructor with out having any parameter is called Default Constructor.

2.PARAMETARISED CONSTRUCTOR:-A Constructor with having at least one value.

3. STATIC CONSTRUCTOR:-To initialize static members of class.

4. PRIVATE CONSTRUCTOR:-

• If we make any constructor as a private we cant create a derived class of it.

• We cant create an instance of that class.

• This class contains for static members only, we access static members by using class name only, not by using an object of that class.

• When we have only static members.

• We use private constructor to prevent for creating an objects.

• We can have parameters in private constructors but no use of it.

5.COPY CONSTRUCTOR:-

• It should be a parameterized constructor.

• In simple words , copy constructor is constructor which copies a data of one object into another object.

• The constructor which creates an object by copying variables from another is called a copy constructor.

• The purpose of a copy constructor is to initialize a new instance to the values of an existing instance.

**COLLECTIONS**

Group of related objects.

There are two types of collections

* **GENERIC COLLECTIONS**
* **NON GENIRIC COLLECTION**

|  |  |
| --- | --- |
| GENIRIC COLLECTION | NON GENIRIC COLLECTION |
| 1.Type Safe | **1.Not Type Safe** |
| 2.Fixed Length | **2.variable Length** |
| Ex .List, Dictionary | **Ex. Array List, Hash table** |

**DICTIONARY**

* Dictionary is also like a collection.
* Dictionary class is a generic collection of keys and values pairs of data.
* The Dictionary class is defined in the system.
* A generic namespace is a generic class and can store any data type in a form of keys and values.
* Each key must be unique in the collection.

**HASH TABLE**

* Hash table is also a collection of the key value pairs.
* Which are organised on the has code of their respective keys.
* When you are declaring has table you are not specify any datatypes its not generic.

**DELEGATES**

* Delegate is a pointer to function.
* A Delegate is a type that represents references to methods with a particular parameter list and return type.
* Delegates have three steps

**1.DECLARATION**

**2.INTIALIZING DELEGATES**

**3.CALLING THE FUNCTION**

**EXCEPTION HANDLING**

* Exception handling in C# is a mechanism for handling runtime errors. It allows developers to catch exceptions that occur during the execution of their code, and to handle them in a controlled way.
* Exceptions are created and thrown when an error occurs in code. Catching an involves using a try- catch block.
* The try block contains the code that might throw an exception. The catch block contains the code that
* will be executed when an exception is thrown.

**CLASSES**

* **ABSTRACT CLASS**
* **SEALED CLASS**
* **STATIC CLASS**
* **PARTIAL CLASS**

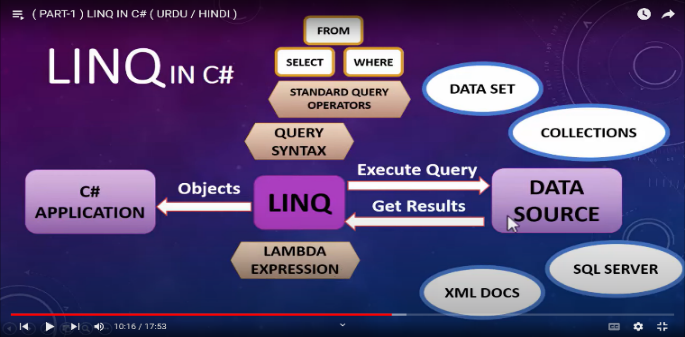
**ABSTRACT CLASS:-**It is a partially defined base class. Partially means half defined class

**SEALED CLASS**:-When we make any class as sealed we can’t inherit that class.

**STATIC CLASS**:-It is a shared member.

**PARTIAL CLASS:-** Partial class means splitting one class into multiple classes.

**LINQ[LANGUAGE INTIGRATED QUERY]**



LINQ introduced in c# version 3.0 as a major part of .net framework 3.5 in 2007.

* LINQ in C# is used to work with data access from data sources such as objects, datasets, SQL server, and xml etc.
* LINQ is a data querying Api with SQL like query syntaxes.
* LINQ allows writing queries even without the knowledge of query languages like sql,xml etc.
* LINQ provides functions to query cached data from all kinds of data sources.
* The data source could be a collection of objects database or xml files.
* Using system. Linq this name space is compulsory when we working with linq statements.
* When we create a console application in visual studio system. linq namespace is added by default.
* If we don’t provide this namespace compile time error will occur.
* 4 standard query operators.

SELECT:-Select \* from my table where age > 20

FROM:-From i in age where i > 20 select i

WHERE:- Select \* from my table where age > 20 order by asc/desc

ORDERBY:- From I in age where i > 20 order by select asc/desc

It is used to work with data access from different data sources like.

SQL,XML DOCUMENTS, COLLECTIONS,DATASETS etc..

**ADVANTAGES**

* Compile time syntax checking.
* It allows you to query collections like arrays enumerable classes etc.
* It increases the readability of the code.
* Query can be reused.
* It gives type checking of the object at compile time.
* It provides intellisense for generic collections.
* It can be used with arrays or collections.
* LINQ supports filtering, sorting, ordering , grouping.
* We can easily convert one datatype into another data type.

**SQL TOPICS**

**INTRODUCTION**

* SQL Stands for STRUCTURED QUERY LANGUAGE.
* It stores sum data base for developing the applications.

**RDBMS**

* RDBMS stands for RELATIONAL DATA BASE MANAGEMENT SYSTEM.
* RDBMS is the basic for SQL and all modern database.
* The data in RDBMS is stored in database objects called tables.
* A table is a collection of related data entries and it consists of columns and rows.

**SQL COMMANDS**

* **­­­­DDL**
* **DML**
* **DQL**
* **DCL**
* **TCL**

**DDL**:-DDL stand for DATA DEFINITION LANGUAGE It has four types.

* **CREATE** :- It is used to create a new table in the database.
* **DROP**:- It is used delete entire table and schema.
* **TRUNCATE:-**It is used to remove all data in table except schema.
* **ALTER:-**It is used to add, delete and modify the new columns or existing column.

**DML:-**DML stands for DATA MANIPULATION LANGUAGE It has three types.

* **INSERT**:-It is used to insert the values in the rows.
* **UPDATE**:-It is used to update the value of a column in the table.
* **DELETE:-**It is used to delete one row in that table.

**DQL:-**DQL stands for DATA QUERY LANGUAGE.

* **SELECT:-**It is used to select the tribute based on the condition describes by where clause.

**DIFFRENCES BETWEEN NORMALISATION & DE NORMALISATION**

|  |  |
| --- | --- |
| NORMALISATION | DE NORMALISATION |
| 1.Keeping the data one table into multiple tables. | **1.Keeping all the data in one table.** |

**JOINS**

**JOINS are three types.**

* **INNER JOINS**
* **OUTER JOINS**
* **SELF JOIN**

**INNER JOINS:-**Combine records from two tables whenever there are matching values in a field common to both tables.

**OUTER JOINS:-**That return matched values and unmatched values from either or both tables.

**SELF JOINS**:-Joining the table by itself only. Joining the same table.

**ORDER BY**

* The order by key word is used to sort the result set in ascending in descending order.
* The records in ascending order by default. To sort the records in descending order use the ““desc ” key word.

**GROUP BY**

* The group by statement groups rows that have the same value into summary rows.

Ex:-find the number of customers in each country.

* The group by statement is often used with aggregate function(count(),max(),sum(),av
* g()) to group the result set by one or more columns.
* Whenever we are using group by you cannot use the where condition. When ever you are using group you should be using having instead of where.

**IN**

* The IN operator allows you to specify multiple values in a where clause.

**EXISTS**

* The EXIST operator is used to test for the existence of any record in a subquery.
* The exist operator returns true if the subquery returns one or more records.

**KEYS**

**PRIMARYKEY:-**Duplicate values and null value not allowed in primary key.

* We have only one primary key in that table.
* Primary key on a column uniquely identifies the records in the table.

**FOREIGN KEY:-**Duplicate values and null values allowed in foreign key.

* Foreign key field in a table in a table and it creates relationship between two tables.

**COMPOSITE KEY:-**

1)Two columns Becomes one key.

2)TO maintain the uniqueness in one table.

3) If we make only one primary key for two columns at a time then we call it as composite key.

**UNIQUE KEY:-**There can be multiple unique keys in the table.

* Can accept one null value.
* Uniquely determines a row which isn’t primary key.

**STORE PROCEDURE**

* Once you can create a store procedure you can use the particular store procedure for en number of applications. Its mean reusability.
* Store procedures are pre compilation statements.
* We don’t make the changes in dotnet code you can make the changes directly in the database.
* We can use the store procedure directly we can go to the database and we can change store procedure.

**FUNCTIONS**

* Functions always we using for reusability.
* Functions in SQL server we actually use calculate something.
* Always functions will be used for select statement.

**DIFFERENCES BETWEEN STORE PROCEDURE & FUNCTIONS**

|  |  |
| --- | --- |
| FUNCTIONS | STORE PROCEDURE |
| 1.Can return only one a value. | **1.Can return 1 or more values.** |
| 2.Only input parameters. | **2.Both input &Output parameters.** |
| 3.Functions cannot call a store procedure. | **3.Store procedure can call a function.** |
| 4.Does not support exception handling. | **4.Support exception handling.** |
| 5.we can only select statement. | **5.we can use insert , update, delete statements.** |

**TIGGERS**

* A Trigger is a special kind of store procedure that automatically executes when an event occurs in the database server.
* Whatever the data is updated or modified by the user then that data will be added to its own table at the same time that data will be stored in the another new table also.
* After inserting trigger, before inserting trigger, and instead trigger.

**There are three types of Triggers.**

* **DML**
* **DDL**
* **LOGON**

**INDEXES**

* When ever we create a primary key the clustered index will be created.
* When ever we are using a where condition with this that particular columns how exactly it picks of these using the indexes.
* First of all it checks it there any indexes available in this particular columns.
* When it creates index how exactly it creates the physical order will be changed when you are creating a clustered index.

**THERE ARE TWO TYPES OF INDEXES**

* **CLUSTRED INDEXES**
* **NON CLUSERED INDEX.**

**1. CLUSTRED INDEXES**

The clustered indexes data will be stored into the database in a physical order.When ever we create a primary key the clustered index will be created.

**2. NON CLUSERED INDEX**.

The non clustered index we can have multiples. It will be Zigzag. Logical order.

**DISTINCT**

If there is any duplicate data in the selected table or query, then the distinct is used to find the list of duplicate data and helps to return the unique data.

**TOP – KEYWORD**

If there are multiple rows in a table, we wish to display only Top 5 or Top 6 rows. Then we can use top keyword.

**ROW NUMBER**

This key word is used to set the row number in that table.

**DIFFERENCES BETWEEN STORE PROCEDURE & TRIGGERS**

|  |  |
| --- | --- |
| STORE PROCEDURE | TRIGGERS |
| 1.Set of statements, which has to be explicitly called by the programmer. | **1.Set of SQL statements which is automatically called when an event occurs.** |
| 2.Can accept parameters. | **2.Cannot accept parameters.** |
| 3.Can return a value. | **3.Cannot return a value.** |
| 4.Can use transaction statement. | **4.Cannot use transaction statements.** |

**NTH VALUE:-**You want to wish highest salary in that table.

SELECT COUNT( COLUMN\_ NAME) FROM table\_name;

**DIFFERENCES BETWEEN WHERE CLAUSE AND HAVING CLAUSE**

|  |  |
| --- | --- |
| WHERE | HAVING |
| It is used to filter data directly on the table. | **It is used to filter the data of groups created on a table.** |
| It is applied as a row operation. | **It is applied as a column operation.** |
| WHERE clause pulls only the filter data based on condition. | **The HAVING clause fetches all data before applying the filter The condition** |
| We cannot use aggregate functions on the WHERE condition. | **It is used on aggregate function on which group has performed.** |
| WHERE works with SELECT, UPDATE, DELETE statement | **The HAVING works only with select statements.** |
| WHERE doesn’t need a GROUP BY clause. It can execute with or without GROUP BY. | **Using HAVING GROUP BY is compulsory.** |

***ADO.NET***

***[ACTIVE DATA OBJECTS]***

**INTRODUCTION**

* ADO.NET is a set of classes (a framework) to interact with data sources such as database and xml files.
* It allows us to connect to underlying data or databases.
* It has classes and methods to retrieve and manipulate data.

**Various configuration Architecture**

**CONNECTED ARCHITECTURE**:-The application remains connected with the database throughout the processing.

**DISCONNECTED ARCHITECTURE**:-The application automatically connects/disconnects during the processing. The application uses temporary data on the application side called a dataset.

**IMPORTANT CLASSES IN ADO.NET**

* **CONNECTION CLASS**
* **COMMAND CLASS**
* **DATA ADAPTER CLASS**
* **DATA READER CLASS**
* **DATA SET CLASS**

**1.CONNECTION CLASS**:-In ADO.net we can use these connection classes to the database .These classes also manage transactions and connection.

**2.COMMAND CLASS**:-The command class provides methods for storing and executing SQL statements and store procedure.

**COMMAND CLASSES ARE FOUR TYPES**

* **EXECUTE READER**:-Returns data to the clint as rows.
* **EXECUTE NONQUERY:-**Executes a command that changes the data in the database, such as update, delete or insert statement or a store procedure that contains one or more of these statements.
* **EXECUTE SCALER**:-This method only returns a single value.
* **EXECUTE XML READER**:-Returns an xml files.

**3.DATA ADAPTER CLASS:-** The Data Adapter is used to connect databases. The Data Adapter is most useful when using data-bound controls in windows forms.

**4.DATA READER CLASS:-**The data reader is used to retrieve data. It is used in conjunction with the command class to execute an SQL select statement and then access the returned rows.

**5.DATA SET CLASS:-** The dataset is the heart of ADO.NET. The dataset is essentially a collection of data table object.

**ASP.NET**

**[ACTIVE SERVER PAGE]**

**INTRODUCTION**

* ASP.NET is a tool for developing web applications in dotnet.
* Web applications are shared applications. Means multiple users can be use at a time.
* The web applications are run by the IIS.
* In this asp.net we can use HTML, CSS, JAVASCRIPT and C# for developing and designing the web applications.
* Browser only can understand HTML. Every data will be converted into html.
* We cannot converted to html the browser cannot display the data.

**IIS**

Internet information services in the particular IIS we need to deploy our web applications.

**VERTUAL DIRECTORY**

* In this IIS we need to create virtual directory for web application.
* Then inside that particular virtual directory our projects files will be copied.

**POST BACK**

Post back happens when data is sent to the server.

We are sending request to the server.

These post back has two types.

* IS POSTBACK
* AUTO BACK

1. **IS POST BACK:-**Is post back helps you to identify if post back has happened or not.
2. **AUTO POST BACK**:-Auto post back helps to you post as soon as control data changes on the control.

**VIEW STATE**

* View state is available only with in that web form.
* View state is used by all asp.net controls to retain their state across post back.

**SESSION STATE**

* Session state variables are available across all pages but only for a given a single session.
* Session state variables are stored on the web server.

**APLICATION STATE**

* Application state variables are available across all pages and across all sessions.
* Application state variables are stored in the web browser.

**SESSION ID**

* The session id is used to uniquely identify a browser with session data on the server.
* The session data is randomly generated by asp.net and stored in a non expiring session cookies in the browser.
* The session id value is then sent in a cookie with each request to the asp.net application.

**RESPONSE.REDIRECT**

* We are using response. redirect it is coming back to a browser when it goes back to the server and it executes the second page.
* The response. redirect method transfer control the specified page and the control is never passed back to the calling page after execution .
* It executes slower.

**SERVER.TRANSFER**

* The server. Transfer method stops the current page from executing and runs the content on the specified page when the execution is complete the control is passed back to the calling page.
* There are only in IIS it is redirect into some other page.
* It is not coming to the browser and goes back.
* It executes faster.

**DIFFERENCES BETWEEN ASP.NET AND MVC**

|  |  |
| --- | --- |
| ASP.NET | MVC |
| 1.view based. [page based]. | **1.Action based.** |
| 2.Return type is always html. | **2.HTML is optional.** |
| 3.Unit testing is difficult. | **3.Unit testing is easy.** |
| 4.Routing is not available.  [we can have our own URL]. | **4.works on routing.** |

**MVC**

**[MODEL VIEW CONTROLLER]**

**INTRODUCTION**

* It is a software architectural pattern.
* In MVC we can develop web applications.

**MODEL**

* It is Just like a class.
* It provides data to view.
* It is optional in MVC.

**VIEW**

* Responsible for look and feel.
* VIEW is actually optional.

**CONTROLLER**

* It is just like a simple class.
* It is heart of MVC.
* In a controller we can have action.

**JSON**

* JAVA SCRIPT OBJECT NOTICE.
* JSON is a simple data exchange format.
* JSON format starts with {} brackets and ends with {} brackets.

**BOUNDLE CONFIG**

It collects data files and keeps into one particular file and it will part of our package.

**BOOTSTRAP**

* BOOTSTRAP is free open source.
* BOOTSTRAP is a popular web framework which is used to create responsive web application that can run even on the mobile device.

**VIEWBAG**

* View Bag is a way to pass data from the controller to the view.
* It is a type object and is a dynamic property under the controller base class.
* View Bag is a dynamic container.
* View Bag doesn’t require type casting for complex data type.

**VIEW DATA**

* In MVC when we want to transfer the data from the controller to, we use view data.
* It is dictionary type that stores the data internally.

**LAYOUT**

* It is the same as defining the master pages but MVC provides some more functionalities.
* MVC allows using multiple layout in a single application.
* We can have multiple views and use only one layout to few view pages.

**SHARED TEST**

* View is same but data need not to be same.
* We need to perform different data from different actions.

**VTEW START**

* \_View start cshtml page is a special view page containing the statement declaration to include the layout page.
* To define the common layout page for all the views you add to your project.

**DIFFERENCES BET WEEB HTML.PARTIAL & HTML.RENDER PARTIAL**

**Both html. Partial and html. Render partial can be used to access or display a partial view in a view.**

|  |  |
| --- | --- |
| HTML.PARTIAL | HTML.RENDER PARTIAL |
| 1.Method that returns mvc html string | **1.Method without any return value it means it returns void.** |
| 2.Partial view result can be stored in string variable. | **2.Rendered partial view result cannot be stored in string variable.** |
| 3.Slow in access. | **3.Fast in access.** |

**ROUTE.CONFIGH**

WE are using Route we can write our own URL.

**ATTRIBUTE ROUTING**

On top of our action itself we can mention the route.

**REGISTER ROUTING**

Register routes means collecting the routes we can have multiple routes inside the register routes.

**HTML HELPERS**

HTML HELPER can be considered as a method that returns you a string.

**STANDARD HTML HELPERS**

**@Html.TestBox ->one Line**

**@Html.Password ->….**

**@Html.TextArea ->Multiple Lines we can add**

**@Html.CheckBox ->Multiple options**

**@Html.RadioButton ->any one of them they would select**

**@Html.DropDownList ->In a List any one They would** **select**

**@Html.ListBox ->Multiple hosts**

**@Html.Hidden ->We can store something in that particular control**

**@Html.Display ->Just display the data**

**@Html.Editor ->Based on model which we are passing it**

**@Html.ActionLink->By action should called when we are click in**

**@Html.BeginForm -> starting tag**

**@Html.Label**

**ENTITY FRAMEWORK**

**INTRODUCTION**

* Entity framework is an object-relational mapper that enables .Net developers to work with relational data using domain specific objects.
* Entity framework is an ORM framework for .Net
* ORM-Stands for Object Relational Mapping.
* ORM means the mapping between object and relational database is called ORM.

**ORM FRAMEWORK IN ASP.NET**

ORM framework automatically creates classes based on data base tables and it can also automatically generate the necessary SQL to create database tables based on classes.

Classes based on database tables

Means first create classes and properties based on those classes on orm automatically it generates.

Database tables-based classes

First create tables on database and then it on ORM Framework automatically it creates classes and properties in Entity Framework.

Advantages

1)It provides auto generated code.

2)It reduce development time.

3)It enables developers to visually design models and mapping of database.

**CRUD OPERATIONS**

C[Create] Insert

R[Read] Select

U[Update]Update

D[Delete] Delete

**JQUERY**

* JQUERY is a java script library.
* jQuery is a lightweight, ”writeless,do more” java script library.
* The purpose of jQuery is to make it much easier to use java script on our website.
* jQuery takes a lot of common tasks that require many lines of java script code to accomplish, and wraps them into methods that you can call with a single line of code.
* jQuery also simplifies a lot of the complicated things from java script like Ajax calls and Dom manipulation.

The jQuery library contains the following features.

* Html/Dom manipulation.
* CSS manipulation.
* Html event methods.
* Effects and animations.
* Ajax.
* Utilities.
* Firstly we download jQuery.
* The jQuery library is a single Java script file. And we reference it with the Html<script>tag(notice that the<script>tag should be inside the <head> section).
* Ex:-<head>
* <script src=”jQuery-3.6.3.min.js”></script>
* </head>
* If we don’t want to download and host jQuery ourself we can include it from a CDN(Content Delivery Network)
* Ex:-<head>
* <script src=[https://ajax.googleleapis.com/ajax//libs/jQuery /3.6.3/jQuery.min.js](https://ajax.googleleapis.com/ajax//libs/jQuery%20/3.6.3/jQuery.min.js)></script>

**JQURY SYNTAX**

* The jQuery syntax is tailor-mode for selecting Html elements and performing some action on the elements.

**Syntax:-**

**$(Selector).action()**

**JQURY SELECTORS**

* jQuery selectors allow you to select and manipulate Html elements.
* jQuery selectors are used to “find” (or select) Html elements based on their name, id, classes, types, attributes, values, of attributes and much more. It’s based on the existing CSS selectors, and in addition it has some own custom selectors.
* All selectors is jQuery start with the dollar sign and parameters. $().

**JQUERY EVENTS**

* jQuery events are the actions that can be detected by our web application.
* They are used to create dynamic web pages.
* An event shows the exact moment when something happens.

**WEB API**

**[APPLICATION PROGRAMMING INTERFACE]**

**INTRODUCTION**

* It provides interaction between software application.
* The Asp.net web Api framework is used to create HTTP services that may be assessed by a variety of clients.
* Web Api is an excellent foundation for making your data and service available to a wide range of devices…
* Web Api is an appropriate platform for developing Restful applications using the Dotnet framework because it is open source.

Features of web Api

* The Asp.net web Api platform is suitable for creating Restful Services..
* Asp.Net web Api supports different datatypes.
* To interface with the web Api server, the Asp.Net web Api server, the Asp.net web Api framework contains a new http client.

Service:-To Interact with some other application.

Service Layer:- It’s like presentation to service, service to business, business to data access layer.

Restful Services

* HTTP based. [Get, Put, Post, Delete]. HTTP has some methods.
* Request and respond to be representation.
* If we send one type of request and we expect the same kind of response. If we pass Json we get Json.
* The URL will be the same does not change. Change only the method type. Uniform URL.
* It is Resource based. Ex: -Employee, Chair. It does not use any methods.

**DIFFERENCES BETWEEN DOTNET FRAMEWORK AND DOTNET CORE**

|  |  |
| --- | --- |
| DOTNET FRAMEWORK | DOTNET CORE |
| 1.Available only for windows. | **1.Available for windows macOS, and Linux.** |
| 2.Dotnet is a platform dependent. | **2.Dotnet core is a platform independent.** |
| 3.Dotnet can support c#,f#,visual basic language. | **3.Doenet core can support c#, f#.** |
| 4.Can create Webapi,Webui,MVC,Signalr. | **4.Can create webforms, Webapi, MVC, Webpages.** |
| 5.Good performance. | **5.High performance.** |
| 6.Compiles after clicking on compile. | **6.Compiles automatically.** |
| 7.Dotnet framework is paid. | **7.Free to use.** |
| 8.hosting:-.net only support its hosting. | **7.kestrel,IIS,Nginx.** |

**POSTMAN**

* The postman is very is useful application for Api Testing.
* The postman is a HTTP client that is used to tests HTTP requests.
* We can utilize Api in our GUI.
* By using postman we can obtain different types of responses comes from web Api.
* It is very useful to validate the requests of webapi.
* Postman’s features simplify each step of building an Api.
* By using postman we can create better Api s and test it faster.
* The postman tool allows you to design ,test,debug,automated testing,document,monitor and publish the Api.
* Postman is also called Api development platform.

**DEPENDENCY INJECTION**

* Dependency injection is achieved using interfaces.
* Interfaces are a powerful tool to use for decoupling.
* Classes can communicate through interfaces rather than other concreate classes.
* Dependency injection is a soft ware pattern.
* Dependency injection is basically providing the objects that an object needs instead of having it construct the objects themselves.
* Dependency injection is a technique where by one object supplies the dependencies of another object.

**SERVICE LIFETIME**

**TRANSIANT:-**A new instance of the service will be created every time it is requested.

**SCOPED:-**These are created once per client request.

**SINGLETON:-**Same instance for the application.

Agile methodology

Sprints planning-2weeks BA analyst

Sprint refinement