C#.NET4.5

NOTES

Mahesh babu

DURGA SOFT

MANOJ ENTERPRISES & XEROX

All soft ware institute materials, spiral-binding,

Printouts & stationery also available ..,

CONTACT: 9963111226

Add: Plot No.40, Gayatri Nagar, Behind HUDA, mithrivannam, HYD.

19/9/2014

2015 New

Why .Net & what is .Net:- 4.5

To understand why .Net, we look into various types of n/w's avialable in the IT field and types of n/w computing's.

Types of Networks:-

(1) Local Area Network (LAN)

(2). Metropoliton Area Network (MAN)

(3). Wide Area Network (WAN)

(4). Enterprise Area Network (EAN)

(1). Local Area Network (LAN):-

Network limited to single organisation located at single place is known as LAN.

Ex:- N/w present in super market, Bank (or) N/w present in college, etc---

(2). Metropolitan Area Network (MAN):-

Network limited to a city and it's near by places is known as MAN.

Ex:- Bank to office, office to supermarket, etc----

(3). Wide Area Network (WAN):-

Network which has no limited in the universe is known as WAN.

Ex:- Internet.

(4). Enterprise Area N/W (EAN):-

These are 2 categories of network

(a). Intranet

(b). Extranet

(a). Intranet:-

Network limited to single organisation located at different places, is known as Intranet

Ex:- Banking N/W.

(b). Extranet:-

Network limited to group of organisation located at different places, is known as Extranet.

Ex:- ATM N/W.

Types of Network Computing's:-

Any type of n/w will be used like LAN(or) MAN(or) WAN(or) EAN, The processing or computing will be performed in any one of the following methods, known as types of n/w computing's.

(1). centralized computing

(2). Client Server computing

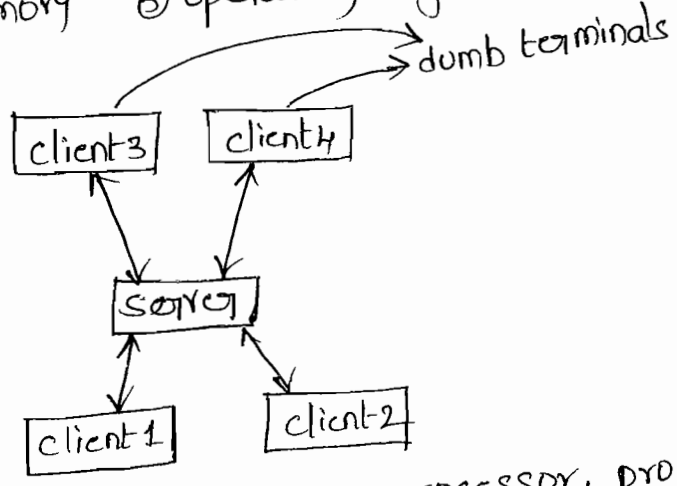
(3). Distributed computing

NO GAP

(ii) centralized computing:-

In this method there is a centrally located Server all other computer's connected to the server is known as client's.

client's in centralized computing are known as dumb terminals. Because client's don't contain any resources like ① processor, ② mother board, ③ Hard disk/ secondary memory, ④ RAM/ primary memory ⑤ operating System.



→ Here client's don't contain any processor, processing will n't be performed at client side, rather processing will be performed at server side

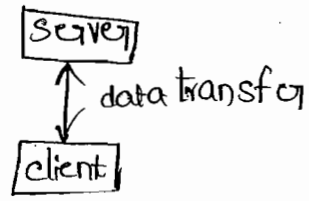
→ Every client will depend on server to execute one or more commands.

Disadvantages in centralized computing:-

↳ The large number of client's burdan on the server will be more, so performance will be decreased.

↳ Data Transfer over the N/W will be increased and the N/W will be slow.

Ex:-



1 command - 2 Transfer
100 " - 100X2
1000 " - 1000X100X2

Advantages of centralized computing:-

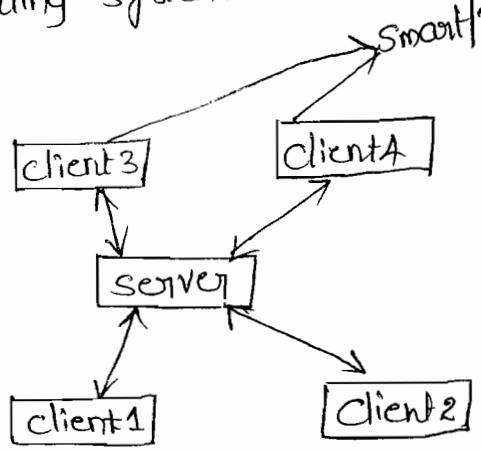
↳ Economically cheaper for the companies.

To overcome the drawbacks of centralized computing we depend on client server computing.

(2), client server computing:-

This is similar to centralized computing, that is centrally located server, computer's connected to this server is known as client's.

But client's are called smart/ Intelligent Terminals because every client will contain all the resources like. ① processor, ② mother board, ③ Hard disk/ Secondary memory ④ RAM/primary memory ⑤ operating system.



→ Here every client will contain's it's own processor, so processing will be performed at client side only.

→ Any client will n't depend on server to Execute one or more commands.

NO GAP

Advantages in Client Server Computing:-

↳ No burden on the server for any number of client's.

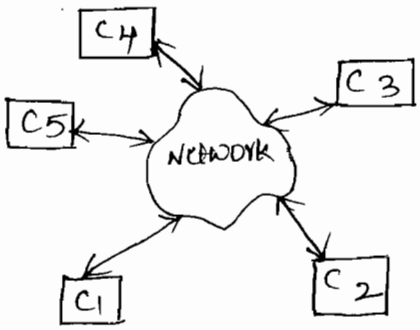
↳ N/W will n't becomes slow.

Disadvantages in client server computing:-

↳ Economically it is burden to the companies maintain client server computing because companies need to purchase all the resources for every client.

(3). Distributed Computing:-

In this method the logic code of the application that is to be executed is dividing among any computer's connected over the network. so that execution can be performed at faster rate.



*→ .Net is the most powerful tool used to build-up the distributed computing applications.

*→ .Net is n't server technology but it supports Server technology

*→ .Net is a frame work tool, which supports many programming languages. .Net supports 63 programming languages.

What is .Net :-

To understand what is .Net we look into various types of software avialable in IT field like

(i). operating system :-

DOS, Windows, Unix, Linux, Solaris, MacOS, OSX, BSD, Andrdid, etc...

(ii). Applications/ packages :-

MS Office, Star office, open office, Tally, wing's, etc...

(iii). Databases :-

DBase, fonpro, MSaccess, SQL server, oracle, mysQL, DBL, Teradata, sybase, etc...

(iv) ERP Applications/ packages :-

ERP → Enterprise Resource planning.

SAP, CRM, Seibel, peoplesoft, oracle APPS, MS Dynamics, etc...

SAP → system's applications products.

→ ERP Application will use large portion or small portion in industry.

(v). Server Technologie's :-

mail server, SMS server, share point, server, biz talk server, CMS, etc....

→ Every server technology has it's own performance

(vi), Testing Tools:- Win runner, Win loader, QTP, selenium, etc...

(vii), programming languages:- Softwares for application development
Fortron, COBOL, PASCAL, C, C++, JAVA, etc...

→ .Net is n't a programming language.
→ .Net is n't an operating system
→ .Net is n't an application/package
→ .Net is n't a database
→ .Net is n't an ERP Application.
→ .Net is n't a server technology.
→ .Net is n't a Testing tool software
→ .Net is a frame work tool, which support's 63 programming language's. In 63 prog. lang's 11 prog. lang's developed by microsoft and 52 prog. lang's are developed by non-microsoft companies.

11 programming language's developed by microsoft:-

1. C#
2. VB
3. J#
4. F#
5. C#
6. JScript
7. Windows power shell
8. Iron python
9. Iron ruby
10. CW.
11. ASML (Abstract State Machine language)

→ ASP. Net is n't a programming language, rather it is server side technology or web technology. ASP. Net code can be written by using any one of the following programming Language's

(i) C# (ii) VB (iii), J#

** Why .Net is n't a programming language :-

.Net is n't a programming language because there is no mechanism or facility to write the code or programming in the .Net.

Rather we write the code in one or more of the 63 prog.lang's supported by .Net and run that code in .Net

What is the meaning of framework:-

In .Net frame work means, the environment where we run the programmes or code is known as frame work.

To check the frame works:-

↳ www.builtwith.com

↳ www.mono-project.com

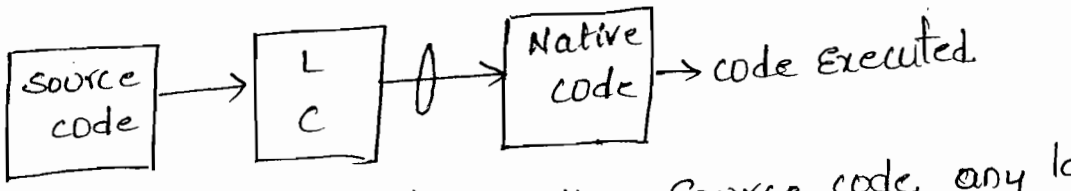
⇒ C#.Net is a most powerful application for android application.

22/9/2024

platform, platform Dependency & platform Independency:-

To understand platform, platform dependency and platform Independency we look into code Execution process in different prog. lang's like COBOL, PASCAL, C, C++, etc...

In these prog lang's source code is compiled by the language compiler, native code is generated, and finally code is executed by the operating system.



At the time of compiling, the source code, any lang. compiler will consider alway's 2 factor's.

(1). process Architecture

(2). operating System Architecture

Native code is generated in such a way that it should be understand by the current processor and current operating system in which code is being compiled.

Platform:-

platform is the combination of processor Architecture and and operating System Architecture

Platform Dependency:-

code generated by the language compiler compilation doesn't run on a different processor and in different

operating system, than which it has been compiled. This nature is known as platform dependency.

Platform Independency:-

If the code generated by language compiler compilation runs on any processor and in any operating system than which it has been compiled, than it is known as platform independent.

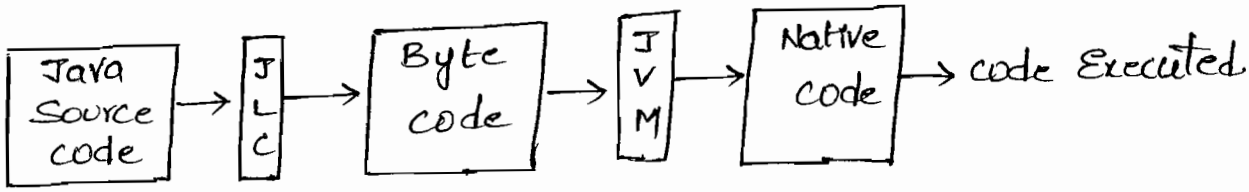
All the prog. lang's like C, C++, COBOL, PASCAL, FORTRON, etc.. were platform dependent.

To make the prog. lang's as platform independent many researcher's suggested that, not to generate native code directly from the language compiler compilation, rather to generate some intermediate code. Based on this suggetion Sun company developed Java and made Java as platform Independent.

Code Execution in JAVA:-

In JAVA source code is compiled by the JAVA lang. compiler and an intermediate code is generated known as "Byte code". This Byte code is platform independent. i.e., we can run on any processor and in any operating system using a special component known as JVM (JAVA VARTUAL MACHINE).

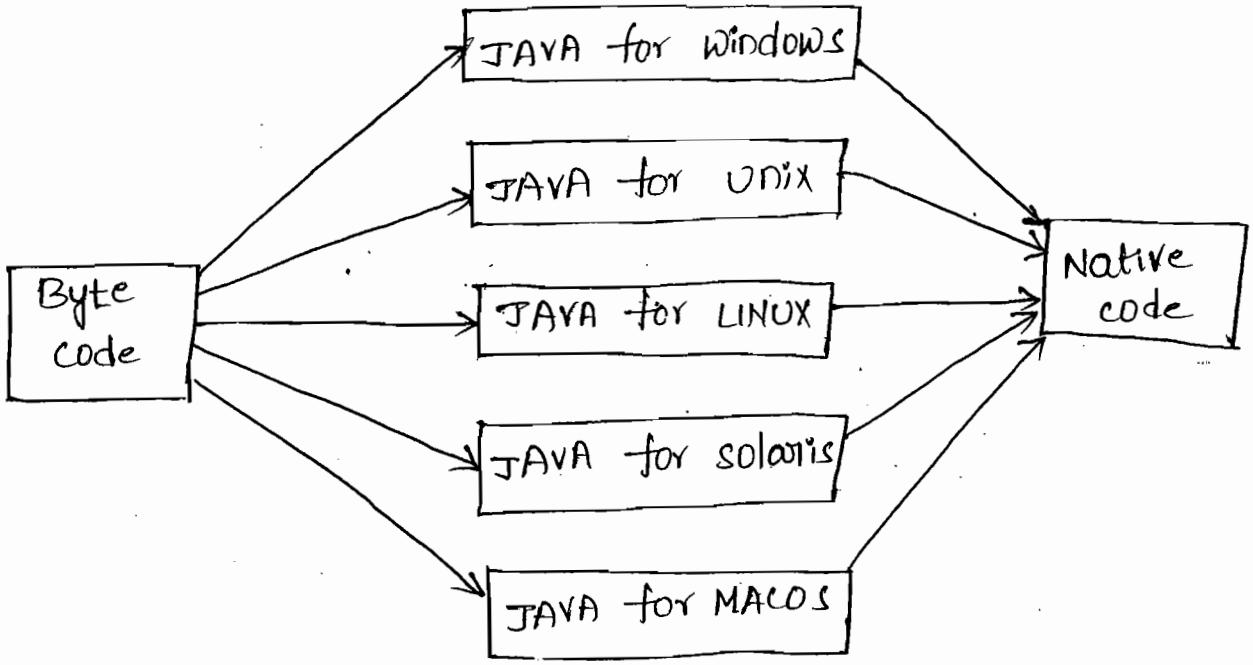
JVM will convert the byte code to Native code, and finally code is executed.



JLC → JAVA Language Compiler

JVM → JAVA Virtual Machine

originally JVM is n't a Hardware component, rather a s/w component. Sun company developed separate JVM for each operating System to make byte code to run in every operating system like



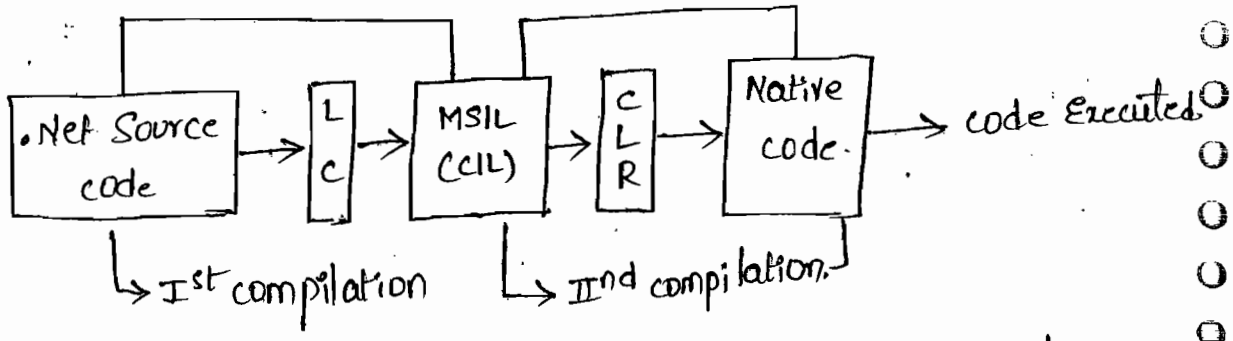
If there is no JVM avialable, for operating system then it is n't possible to run the byte code or JAVA code in that operating system. In future, if any company or person develops any operating system than someone should develop new JVM, in support to only Java code will be Executed.

23/9/2014

Code Execution in .Net:-

In .Net code is compiling 2 times. In first compilation, Source code is compiled by the language compiler and an intermediate code is generated known as MSIL (MICROSOFT INTERMEDIATE LANGUAGE)

In second compilation MSIL code is converted into Native code by the CLR (common language runtime)



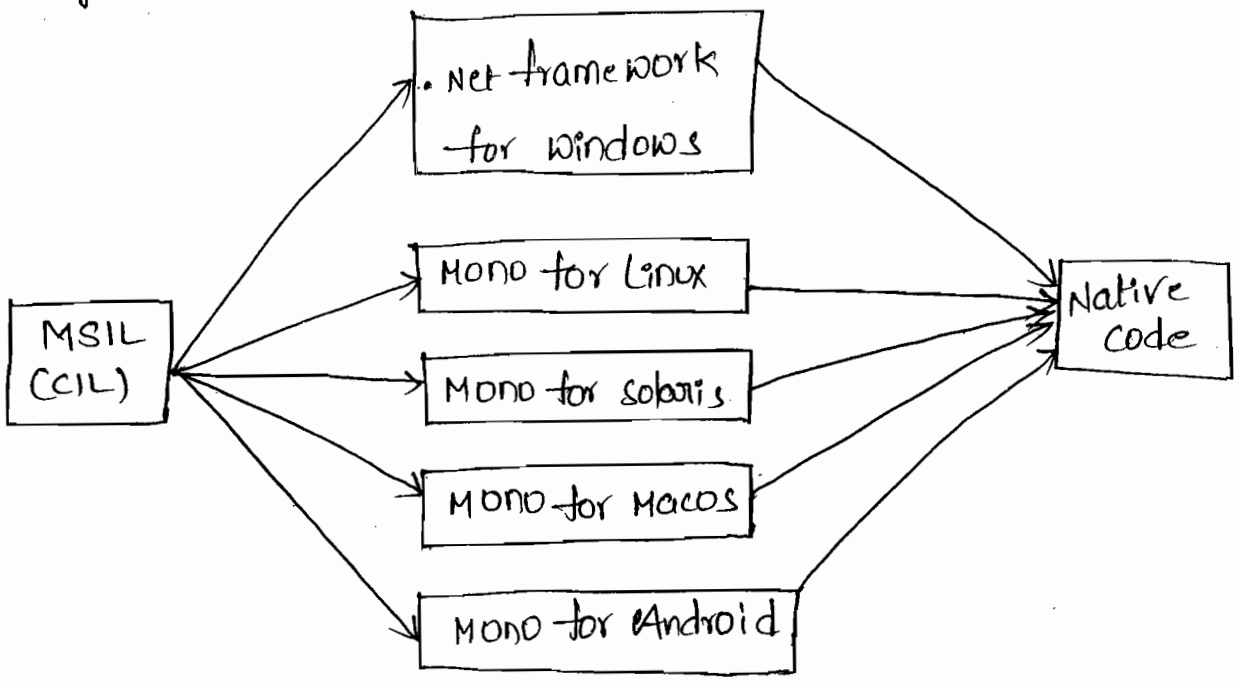
Always in both compilation's first compilation is slow and IInd compilation is faster

Here MSIL code is only processor independent or CPU independent. But n't on OS independent. So MSIL code can be run on any processor that only in windows operating system using .Net frame work. Because microsoft developed .Net frame work only for windows operating system.

.Net framework is n't avialable for Linux, solaris and other operating system's.

To run .Net code or MSIL code in other operating system's we use other software called mono..

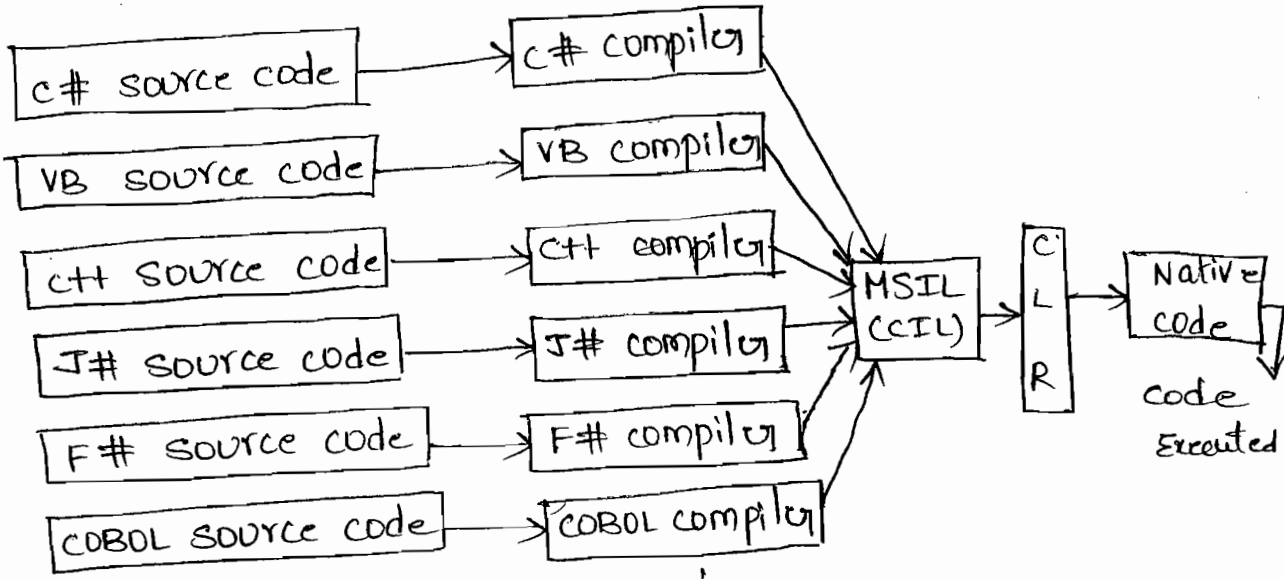
Mono is developed separately for each operating system like,



→ In .Net source code means code the type is written in any one of the programming languages supported by .Net.

→ Among In these 63 prog. lang's every prog. language has its own language compiler.

→ code we may write in any one of these prog. lang's compiling by that language compiler, same common form of Intermediate code is generated i.e., MSIL (microsoft Intermediate Language)



Int. Blue
MS Exam 2 to 3 Bl

** Components of .Net frameworks:-

① CLR (common language runtime)

↳ ① CLI (common language Infrastructure)

↳ ① CLS (" " specification

↳ ② CTS (" Type system)

↳ ② GC (Garbage collector)

↳ ③ JIT (Just in Time) compiler

Int. Blue

② BCL (Base class Libraries) (or) FCL (framework class Libraries)

Common Language Specification (CLS):-

Language Specification:-

When we write the code in any prog. lang. we follow some syntatical rules, These syntatical rules that are followed to write that code in any prog. language are known as language specifications.

We know that .Net supports 63 prog. lang's. Among these 63 prog. lang's every prog. lang has its own langauge specification or syntatical rules.

→ If we consider 2 prog. lang's like C# & VB we write the code for IF-ELSE like

C#

```
if (condition)
{
  Statements;
      ⋮
}
else
{
  statements;
      ⋮
}
```

VB

```
If condition Then
    Statements
      ⋮
else
    statements
      ⋮
End If
```

Among these 63 prog. lang's though every prog. lang has it's own lang. specification (or) syntactical rules. one programming language can't understand. other prog lang's, language

specifications or syntactical rules.

But all these prog. lang's, code is commonly run by the CLR. This is because CLR doesn't know or can't understand any of the 63 prog. lang's, language specification or syntactical rules. Rather CLR has its own language specification for MSIL.

All prog. languge, language compiler will genuate MSIL from their source code by following the Language specification of CLR for MSIL. SO CLR is understanding MSIL.

This lang. specification of CLR for MSIL is common for all the 63 prog. lang's supported by .Net. SO, this is known as common language specification (CLS)

24/9/14

CLS is responsible to provide language Interoperability.

Defnition of language Interoperability:-

providing code executions about, that has be written in other prog. lang's, is known as language Interoperability.

In .Net lang. Interoperability is achieved in 2 ways.

(1). managed code.

(2). unmanaged code.

11). Managed code:-

code for which MSIL form is available, after the language compiler compilation, is directly run by the CLR. This is known as managed code.

For managed code CLR will provide all the features and facilities like,

(1). Language Interoparability
(2). common Type system
(3). Automatic memory management
(4). JIT compilation
(5). code Access Security
(6). Eception handling mechanism, etc...

In simple tam's to say for managed code, CLR is managed every thing.

Eg: of managed code:-

63 prog. lang's code supported by .NET

12). Un managed Code:-

code which is developed before .Net for which MSIL form is n't available is n't executed by the CLR, directly, rather CLR will redirect to this operating system and OS will run the code and will give result to CLR, This is known as unmanaged code.

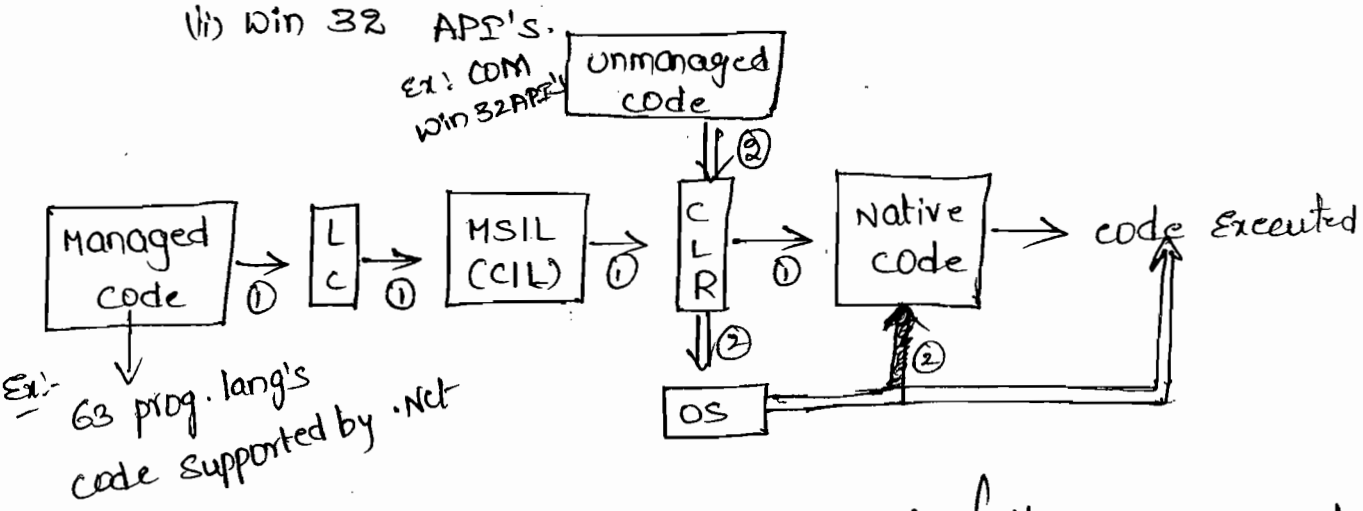
FOR unmanaged code CLR doesn't provide any facilities or features of .Net.

In simple term to say CLR will n't manage any thing for unmanaged code.

Eg. of unmanaged code:-

(i) common component

(ii) Win 32 API's.



Always managed code Execution is faster, unmanaged code Execution is slow.

(2). common Type System (CTS):-

We know that, .Net supports 63 prog lang's in managed code Execution.

Among these 63 prog. lang's every prog. language has its own datatype System.

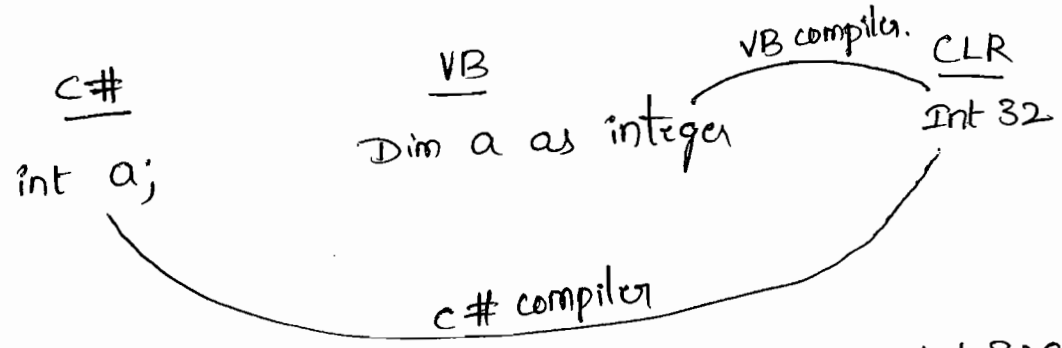
If we consider c# and VB programming languages, we create an integer variable like

c#	VB
int a;	Dim as a integer.

Though every prog. language has its own datatype System, one prog. language can't understand other prog. lang's datatypes.

But all these prog. lang's datatypes are commonly run by the CLR. This is because CLR doesn't know or can't understand any of the 63 prog. lang's datatypes. rather CLR has its own datatype system for MSIL.

All the prog. language datatypes are converted into CLR's datatype by the language compiler's.



Here c# compiler will compile it's int type to int 32 CLR, Similarly VB compiler also compile it's int type to int32 of CLR. so CLR is understanding it's own datatype system i.e., Int 32

This data type system of CLR is common for all the 63 prog. lang's supported by .Net, so this is known as common type system (CTS)

This CTS of CLR is divided into 2 categories

(1) Value Types
(2) Reference Types.

(i) Value types:-

The datatypes which are capable of storing the data directly into their memory locations are known as value types.

Ex:- int a=10; a [10]

(ii) Reference types:-

The datatypes which can't store the data directly into the memory locations, rather refers to other memory locations. Where data is stored are known as reference types

Ex:- string s = "welcome" [ABIOIF] → (welcome) ABIOIF

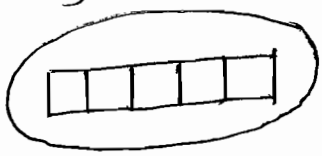
S

25/9/14

Differences b/w values types and reference types:-

Value Types	Reference Types
→ store the data directly into their memory locations.	→ Don't store the data directly into their memory locations, rather refers to other memory location where data is stored.
→ memory is alloted at compile time	→ memory is alloted at run time.

→ memory allocation is made within the stack, i.e, contiguous memory locations



→ memory allocation is made with in the Heap i.e., in Random memory locations.

→ CLR doesn't provide Automatic memory management

→ CLR Provides Automatic memory management.

→ occupies less memory

→ occupies huge memory, a single reference type variable can occupy maximum of upto 2GB memory

→ If data is not initialized, then stores the default value in to the variable

Ex: int a; a [0]

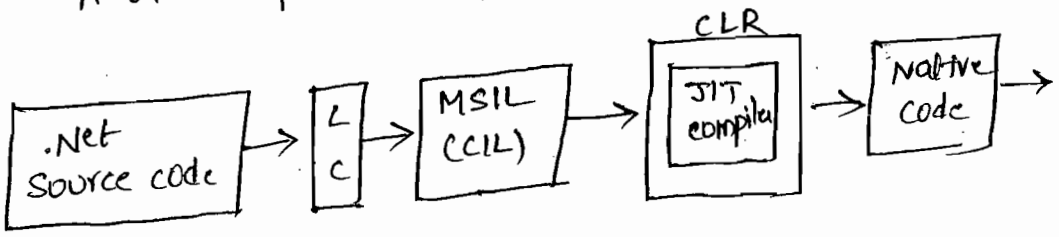
→ If data is not initialized, then stores the null value reference into the variable

Ex: string s; s [null]

(2). JIT (Just In Time) Compilor:-

JIT compilor is responsible to compile the MSIL code and to generate the Native code. where as CLR will run the code by providing all facilities and features of .Net.

A JIT compilor in also known as "JITTER"



Disadvantages Using JIT compiler:-

(1), Native code generated for one program can't be used in other programs.

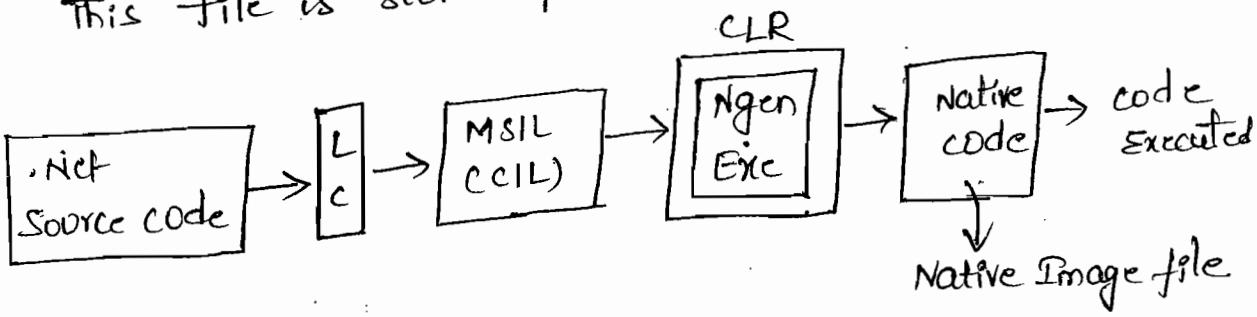
(2) unnecessary compilative will be more. which will reduce application performance.

To overcome these drawbacks we use another compiler tool i.e., Ngen.Exe

Ngen.Exe:-

This compiler tool is used to convert MSIL code and to generate the Native code but this Native code stored into a file. i.e, Native image file.

This file is stored permanently onto the Harddisk (HD).



Advantages Using Ngen.Exe:-

There is no unnecessary compilation, so this will n't reduce the application performance

Native code generated in one program can also be used in other programs.

Int. Blues

* what is the difference b/w Assembly & class Name spaces.

Ans:- Assembly is collection of classes and Name space is also a collection of classes.

→ Assembly is physical entity and Name space is a logical entity. Because Assembly have file name on the disc. and Name space will n't have file on the disc.

→ creating Assembly ~~is compulsary~~, creating name space is optional.

→ .Net Versions

1.0
1.1
2.0 → 2005 → 500+ BCL's, 12,909 classes, 4,01,759, PDF's
3.0
3.5
4.0
4.5
4.5.1 → 6 Lakhs
.2 → 2013

.Net
↳ class Libraries/Assembly
↳ Name spaces
↳ classes
↳ predefined functions
↳ code

→ class Libraries

- User defined C.L's.
- Base class Libraries (or) System defined B.C.L (or) predefined functions.

Location:- C:\windows\Assembly.

→ Assembly will contain many number of classes (or) Name spaces but Name spaces will n't contain Assembly

→ path to find a file in .Net is given below.

Namespace Name . Class Name . predefined ~~file~~ function Name

26/9/2014

Base class Libraries:—

class Library:—

In our daily life we find so many libraries like Book library means collection of books. CD Library means collection of CD's, video library means collection of video's similarly class Library means collection of classes where every class will contain so many predefined functions which are helpful to write the code in .Net.

→ class Libraries in .Net are also known as Assembly's.

→ class Libraries or Assembly's in .Net are very much similar to packages in Java.

→ class Libraries are known as building blocks of the application in .Net

→ class Libraries in .Net are also known as self describing in Nature

microsoft maintains particular hirarchy to the class Libraries. Like

city
↳ Book Libraries
↳ Racks
↳ Books
↳ pages
↳ matter

.Net
↳ class Libraries / Assemblies
↳ Name spaces
↳ pre defined functions
↳ code

This Hirarchy especially used ~~full~~ in following Things.

(1) To provide easy Access to the pre-defined functions.

(2). To avoid dupilcate Naming problems.

To Access any pre-defined functions we use the hierarchy like

Namespace Name . class Name . predefined function Name

member Access operator.

Difference b/w Assembly & Namespace :-

Assembly	Namespace
→ An Assembly is collection of similar group of classes	→ A Namespace is also collection of similar group of classes
→ An Assembly contains one or more namespaces within it.	→ A namespace can never contain Assemblies
→ An Assembly is a physical entity	→ A Namespace is a logical entity
→ creating An Assembly is compulsory	→ creating a Namespace is optional.

A class Library can be divided into 2 categories

(i). User defined class libraries

(ii). Base class libraries.

(i). User defined class Libraries:-

These are created by the programmer for the purpose of reusability within the applications or programs.

(ii) Base class Libraries:-

→ These are created by microsoft.

→ These are installed into system, when we install .Net framework.

From windows 7 onwards framework is avialable with windows operating system only. The physical location of base class libraries is c:\windows\Assembly.

Framework 2.0 version contains 500+ BCL's, 12,909 classes, 4,91,759 predefined functions.

current version of frame work 4.5.1 contains more than 6 lakh's of PDF's.

program:-

set of instructions used to perform required task, stored in secondary memory.

27/9/2014

DLL/Exe:-

Any class Library or Assembly in .Net will exist in the form of DLL/Exe.

DLL:- A DLL is a dynamic link library which is used as a supporty for other applications.

Library functions are linked applications at run time.

A dll doesn't contain any entry point or main method so dll can't run independently.

Exe:-

An exe is an executable file, It self an application n't used as a supporty file. A Exe contains main method or entry point so Exe can't run independently.

Differences b/w DLL & Exe:-

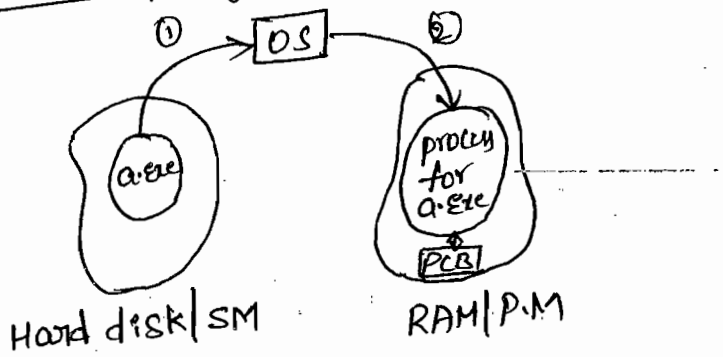
DLL	Exe
→ cannot run Individually	→ Runs Individually.
→ Used as a supporty file for other applications	→ It self an application
→ Doesn't contain Entrypoint (Main function) so can't run individually	→ contains Entrypoint (Main function) so runs individually
→ A program / Application without Main creates a DLL after compilation	→ A program / Application with main creates an Exe after compilation.
→ OS doesn't create a separate process for any DLL rather DLL will run in the same process created for an Exe.	→ OS creates a separate process for each Exe it executes.

Note:- creating a process is an overhead to the OS so DLL doesn't contain main function to reduce the burden on OS.

Interviews ***

Explain how the process will be created in .Net:-

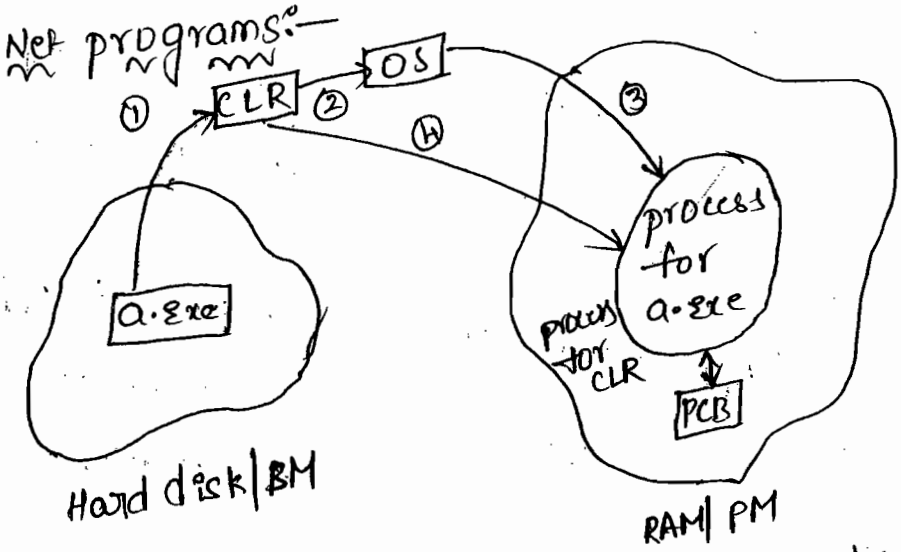
In Non.Net programs:-



Step 1:- A program created in Non.Net starts its execution sends request to OS.

Step 2:- OS will create a process, PCB and maintains everything for this program

In .Net programs:-



Step 1:- A program created in .Net starts it's execution, sends request to CLR

Step 2:- CLR will send Request to OS.

Step 3:- OS will create a process for CLR.

Step 4:- CLR will create a process PCB for a.exe within this process.

In .Net creating a process, PCB and maintaining everything for any .Net program will be looked after by CLR. So CLR will run the program.

Int. Bluz

***

What is Application domain / APP Domain in .Net :-

A process created by the CLR, for any .Net program it run's is known as ~~pro~~ application domain / APP domain.

C# THE PROGRAMMING LANGUAGE

→ We know that, .Net support's 63 prog. lang's.

→ Among these 63 prog. lang's c# is the most powerful prog. lang. Because c# contains all the features of c++, all the features of Java and also additional features,

$$\boxed{C\# = C++ + Java + \text{Additional features.}}$$

Why the Name c# :-

microsoft developed this programming lang. as the successor of both c and c++.

Successor of both c and c++:-

→ so first letter c indicates it is successor of c and c++ prog. lang's.

→ The symbol # should be pronounced as sharp only because microsoft has taken the symbol from music field and not from IT field.

In music field this is known as sharp note and meaning is most powerful.

so overall c# indicates the most powerful prog. lang which is successor of c and c++.

Structure of program in C#:-

```
class  className
{
  static void main( )
  {
    statements;
      |
  }
}
```

{ → opening delimeter

} → closing delimeter

; → Terminator.

Note:-

Note:- windows 7/8 SP1 professional/ultimate [If SP1 not avialable goto microsoft website and type windows 7 SP1 and download it]

Vs.Net 2012

SQL Server

First program in C#:-

Example to print "welcome"

open note pad application, type the code

```
class Example1
{
  static Void Main( )
  {
    System. Console. Write ("welcome");
  }
}
```

C : printf ("welcome")

Java : System.Out.println ("welcome")

C# : System. Console. Write ("welcome)

Namespace Name — ClassName — predefined function Name

→ Save the file with save name as class name that contains main method.

→ within paranthysis Example.cs.

→ Goto command prompt type following:-

Type E:↵

Type CD csharp&AM↵

Type csc Example1.cs↵

At this step, csharp code is compiled by csharp compiler, MSIL code is generated, and this code is stored into .exe file, Because program contains main method.

For second compilation and running of the program, then Type following like

Type Example1.[exe]↵
↳optional.

→ c# supports all the escape sequence characters supported in c and c++

\n → New line

\t → Tab space

\0 → null

\b → Backspace

\\ → backslash

\ → Escape Sequence

Write a program to print your personal details on the Screen:-

```
class Example2
{
 static Void Main( )
  {
   System . Console . WriteLine ("My Name is:-Vasu");
   System . Console . WriteLine ("My Age is:- 21 years");
   System . Console . WriteLine ("My qualification is : B-Tech");
   System . Console . WriteLine( "my village is:-ONGOLE");
   System . Console . WriteLine( "my country is :- INDIA");
  }
}
```

In the above program Namespace system is repeated in every statement to overcome that, we can include Name space Name in the begin of the program like

```
using System;
class Example2
{
 static void Main( )
  {
   system . console . Writeline ("my Name is :- Vasu");
   :
   :
  }
}
```

Datatypes in C#:-

- Data Types in CTS
  - Reference Types
    - class Types
      - Object
      - String
    - Interface Types
    - Array Types
    - Delegate Types
  - Value Types
    - simple Types
      - signed Integral
        - sbyte
        - short
        - int
        - long
      - unsigned Integral
        - byte
        - ushort
        - uint
        - ulong
      - unicode Character
      - IEEE floating
        - float
        - Double
      - Decimal
      - Boolean
    - Enum Types
    - struct Types

Equivalent Datatypes chart in CLR & C#:-

category	C# data type	classname (CTS Data Type)
Integer	byte	Byte
	sbyte	SByte
	short	Int16
	int	Int32
	long	Int64

	ushort	UInt16
	uint	UInt32
	ulong	UInt64
Floating point	float	Single
	double	Double
Logical	bool	Boolean
other	char	Char
	decimal	Decimal
	IntPtr No built-in type	IntPtr
	UIntPtr No built-in type	UIntPtr
class objects	object	Object
	string	String.

NO GAP

Data types, their sizes and range:-

Type	Range	Size
sbyte	-128 to 127	Signed 8 bit Integer.
byte	0 to 255	Unsigned 8-bit Integer
char	U+0000 to U+ffff	Unicode 16 bit character.
short	-32768 to +32767	Signed 16 bit integer
Ushort	0 to 65,535	Unsigned 16 bit integer
int	-2,147,483,648 to +2,147,483,647	Signed 32 bit integer
unit	0 to 4,294,967,295	Unsigned 32 bit integer
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64 bit integer
ulong	0 to 18,446,744,073,709,551,615	Unsigned 64 bit integer
float	1.5×10^{-45} to 3.4×10^{38}, 7-digit precision	32 Bits.
double	5.0×10^{-324} to 1.7×10^{308}, 15-digit precision	64 Bits.
decimal	1.0×10^{-28} to 7.9×10^{28}, 28-digit precision	128 Bits.

Working with Variables in C# :-

```
int a;
int a, b, c;
int a; float b; string s;
```

```
int a = 10;
int a=10, b=20, c=30;
int a = 10; float b = 20.5f; double c = 40.5; string s = "welcome";
```

Example to create a variable and Example to print the data on the screen to user:-

```
Using System;
class Example3
{
    Static Void main()
    {
        int a=10;
        console.WriteLine ("value of a is:-" +a);
        console.Writeline ("value of a is:-{a}", a);
    }
}
```

concatination operator

concatination method

output stream Argument.

29/9/2014

Example to print more than one variables data on the screen to user:-

```
using System;
class example4
{
    static Void Main()
```

```
{
    int a=10, b=20;
    Console.Writeline("values of a and b are -" +a +"," +b);
    Console.Writeline("values of a and b are - {0} {1}", a, b);
}
}
```

Reading the data from the user:-

To read the data from the user, we use readLine method.

Syntax of ReadLine Method:-

```
variable = Console.ReadLine();
```

int a=10; ✓	String a=10; ✗	double a=10;
int a="10"; ✗	String a="10"; ✓	

int a=10;	String a="10"
int b=20;	int b=20;
int c=a+b; ✓	int c=a+b; ✗

→ Read line will read the data, with string type and the return type also string.

→ Though user enters numbered data, read line will read it as string only.

→ Here in L.H.S side the datatype is equal or grater than to R.H.S side

Example with to print your name on the Screen to User:-

```
using System;
Class Example5
{
  static void Main()
  {
    console.Write("Enter your Name:-");
    String s = Consok.~~Write~~ ReadLine();
    Console.WriteLine("your Name is:-"+s);
  }
}
```

Type conversion methods used to convert the data from String type to Numbered type:-

1). Convert.ToTargetType class ("Data") → Data

Ex:- Convert.ToInt32("10") → 10 → 4 Byte
Convert.ToInt16("10") → 10 → 2 Byte
Convert.ToInt64("10") → 10 → 8 Byte
Convert.ToDouble("10") → 10.0

2). TargetType class.Parse("Data") → Data

Int32.parse("10") → 10 → 4 Byte
Int16.Parse("10") → 10 → 2 Byte
Int64.Parse("10") → 10 → 8 Byte.
Double.Parse("10") → 10.0

13), TargetType.Parse("Data") → Data

int.Parse("10") → 10 → 4 Bytes

short.Parse("10") → 10 → 2 Bytes

long.parse("10") → 10 → 8 Bytes

double.Parse("10") → 10.0

→ In 1st Type the data type already will have in CLR's language so there is no need to convert the datatype from one datatype to another data type

→ In 2nd and 3rd data type conversion the data will be converted from String to Number type. So it makes time taken while at conversion.

→ In 3 data type conversion process, 1st method is easier than other 2 conversion types.

Steps for Basic programming:-

Step 1: Identify and declare variables.

Step 2: Accept the required data from the user.

Step 3: perform the required calculations.

Step 4: print the required the data to the user.

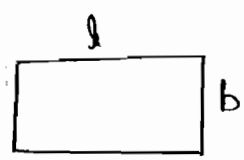
30/1/11

1. write a program to accept any 2 numbers and find sum of the two Numbers.

program:-

```
using System;
class Example6
{ static Void Main( )
  {
    int a, b, c; ---> ①
    Console. Write("Enter any two numbers:-");
    a= Convert. To Int32 (Console. Read Line( ));
    b= Convert. To Int32 (Console. Read Line( ));  } ②
    c=a+b; ---> ③
    Console. Writeline ("sum is:-" +c); ---> ④
  }
}
```

2. write a program to accept length & bredth of a rectangle find Area, perimeter and print on the Screen to the user.



A = l * b

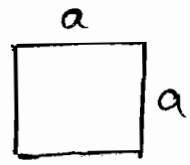
P = 2 * (l+b)

program:-

```
using System;
class Example7
{ Static Void Main( )
  {
    int l, b, A, P;
    console. Write("Entu the values of l & b:-");
    l = convert. To Int32 (Console. Readline( ));
    b = convert. To Int32 (Console. Readline( ));
    A = l * b;
    P = 2*(l+b);
```

```
console. WriteLine ("Area is :- " +A);
Console . WriteLine ("perimeter is :-" +P);
}  console . Write (OR) Line ("Area and perimeter of rectangle are:-"
                                  +A +";" +P);
}
```

3. write a program to accept side value of a square, calculate Area perimeter to print on the ~~Screen~~ to the User

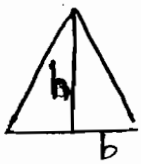


$A = a^2 = a*a$

$P = 4*a$

```
program:- Using System;
    Class Example 8
    {
      static Void Main( )
      {
        int  a, b, c;
        console. Write ("Enter the value of a:- ");
        a = convert. ToInt32 (Console. ~~Write~~ ReadLine ());
        b = a* a;
        c = 4* a;
        console. WriteLine ("Area and perimeter of a square are:-"
                                  +b +";" +c);
      }
    }
```

4. write a program to accept ~~to~~ Base and height of a triangle, calculate Area to print on the Screen to User.

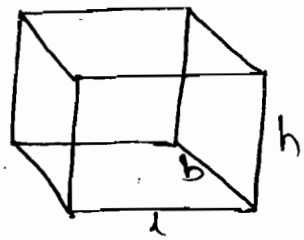


$A = \frac{1}{2} * b * h.$

```
program:- Using System;
    class Example 9
    {
      static Void Main( )
```

```
{ int   b, h, A;
  console.Write ("Enter the values of b and h :-");
  b = convert.ToInt32 (console.ReadLine());
  h = convert.ToInt32 (console.ReadLine());
  A = 1/2 * b * h;
  console.WriteLine ("Area of the triangle :-" +A);
}
}
```

5. Write a program to accept length, bredth and height of a cuboid calculate ~~value~~ volume, total surface area, To print on the Screen to use.



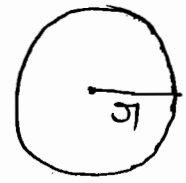
$V = l*b*h;$

$TSA = 2*((l*b)+(b*h)+(h*l))$

program:-

```
using System;
class Example 10
{
  Static Void Main()
  {
    int l, b, h, V, A;
    console.Write ("Enter the values of l and b and h:-");
    l = convert.ToInt32 (console.ReadLine());
    b = convert.ToInt32 (console.ReadLine());
    h = convert.ToInt32 (console.ReadLine());
    V = l * b * h;
    A = 2* ((l*b)+(b*h)+(h*l));
    console.WriteLine("volume and Total Surface
                Area of cuboid are:- "+V")" +A);
  }
}
```

6. write a program to accept radius of a circle, calculate Area, perimeter print on the Screen to the User



$A = \pi r^2$

$= 3.14 * r * r$

$= Math.PI * r * r$

$P = 2 * 3.14 * r$

$2 * Math.PI * r$

Program:-

```
Using System;
class Example11.
{ static Void Main()
  {
    double r, A, P;
console. Write("Enter the value of r is:-");
    r = Convert. ToDouble (Console. ReadLine ());
    A = Math. PI * r * r.
    P = 2 * Math. PI * r.
Console. WriteLine(" Area and perimeter of the Circle
                                are:-" +A +")"+P);
  }
}
```

7. write a program to accept any two number's from the User find first number to the power of second Number and print on the Screen to User.

Program:-

```
Using System;
class Example12
{ Static Void Main()
  {
    int a, b; double Result;
    Console. Write("Enter the values of a and b are:-");
    a = Convert. ToInt32 ( Console. ReadLine());
```

```
b= Convert.ToInt32 (Console.ReadLine());
Result = Math.Pow(b); (or) Math.Pow(a, b) "{0} to the power of {1} is
Console.WriteLine(" first Number {0} to the power of second
:-{2}", a, b, Result);
}
}
```

8. write a program to accept the temperature in centigrade and convert to Faraheet and print on the Screen to User.

program:-

```
Using System;
class Example13
{ static Void Main()
  { int A, C, F;
    Console.Write ("Enter the value of C:-");
    C = convert.ToInt32 (Console.ReadLine());
    A = C*9/5;
    F = A+32;
    Console.Write ("Faraheet temperature is:-"+F);
  }
}
```

9. write a program to accept temperature in Faranheet convert to centigrade and print on the Screen to User.

program:-

```
Using System;
class Example14
{static Void Main()
  { int A, C, F;
    Consol.Write ("Enter the Value of F:-");
    C= Convert.Int32 (Console.ReadLine());
```

```
A= F-32;
c= A* 5/9
console.Write("centigrade temperature is:-"+c);
}
}
```

10. write a program to accept any number from the user find square value, cube value and print on the Screen to user

cube:- Math.Square (3) 27,0.3)
sqrt:- Math.power (9,0.5); 3
Math.POW(a,b)
Math.square(3)
Math.power (a,3)

program:-

```
Using System;
class Example15
{ static Void Main()
  { int a, b, c;
    Console.Write ("Enter the value of a:-");
    Convert.ToInt32 (Console.ReadLine());
    b = a* a;
    c = a* a* a;
    Console.Write("Enter the square value, cube value are:-"
                                    +b+";" +c);
  }
}
```

11. write a program to accept any number from the user find it's square root value, cube root value and print on the Screen to user

a = 27
$\sqrt[3]{27} = 27^{1/3}$
$= 27^{0.33}$

```
Using System;
Class Example16
{ Static Void Main()
  { int a; double R;
    Console.Writeline("Enter any Number:-");
```

```
a = C.ToInt32(C.R.Line());
R = Math.pow(a,0.33);
C.W.Line("cuberoot value of {0} is {1}",a, R);
}
}
```

01/10/2019

12. Write a program to accept principal, time period, rate of Interest then calculate Interest Amount and Total Amount using Simple Interest

$I = (P*T*R)/100$

$A = P+I.$

program:-

```
Using System;
class Example17
{ static void Main()
  { int P, T, R, A, B, C;
  Console.Write("Enter the values of principle, time period,
                rate of interest:-");
  P = Convert.ToInt32(Console.ReadLine());
  T = Convert.ToInt32(Console.ReadLine());
  R = Convert.ToInt32(Console.ReadLine());
  I = (P*T*R)/100;
  A = P+I;
  Console.WriteLine("The Interest amount and Total Amount
                are" + I +")" + A);
}
```

13. Write a program to accept bookname, Author name, publisher name and M.R.P of the book, and discount % of the book. calculate discount amount and selling price of the book print on the Screen to User.

MRP } → Double
DA }

DA = MRPXDA/100
SP = MRP−DA

Program:-

```
Using System;
class Book
{ static Void Main()
  {
    double = M, D
    String = BName, AName, PName;
    Console.Write("Enter the values of BName, AName, PName M and D:-");
    M = Convert.ToDouble(Console.ReadLine());
    D = Convert.ToDouble(Console.ReadLine());

    BName =
```

How to open Command prompt:-

1. open windows explorer

→ double click on windows folder

→ Double click on Microsoft.NET

→ Double click on Framework

→ Double " " version folder.

c:\windows\Microsoft.NET\Framework\v4.0.30319

→ copy this total path from address bar and select computer

→ by clicking left button

→ select properties

→ click on advanced Setting

→ Click on Environment variable

→ click on New

→ Type variable name as "path"

→ paste the variable value copied from address bar or

Type

c:\windows.

→ click on ok 3 times. Then Execute the program.

14. There is a wall in the form of rectangle, write a program to accept length and bredth of the wall. calculate cost to paint the wall at the rate of 15Rs/sqm.

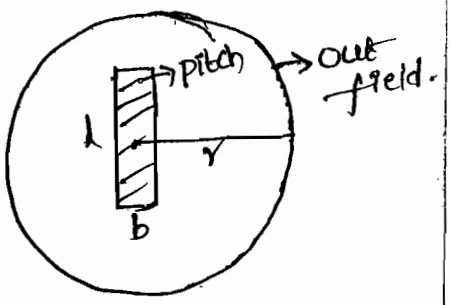
program:- Using System;

```
class wall
{
 static void Main()
 {
  int l, b, A, c;
  Console.Write("Enter the length and bredth of wall:-");
  l = Convert.ToInt32(Console.ReadLine());
  b = Convert.ToInt32(Console.ReadLine());
  A = l*b;
  c = A* 15;
  Console.Writeline("The cost for painting the
                                wall is:- " +c);
 }
}
```

15. A Farmer has a garden in the form of a rectangle, Farmer would like to construct a path around the garden with 2m wide. write a program to accept length and bredth of the field. calculate cost to construct the path at the rate of 20 rupees/sqm.

16. A Farmer has a garden in the form of a circle. Farmer would like to construction a fencing around the garden. Write a program to accept radius of the garden, calculate cost to construct the fencing at the rate of 10 Rs/m.

17. There is a cricket ground in the form of circle. Management would like to construct a pitch in the ground. Write a program to accept radius of ground, length and breadth of the pitch. calculate cost to construct the pitch at the rate of 25 Rs/sqm. Also find the cost to construct the outfield at the rate of 50 Rs/sqm.



8/10/2014

OPERATORS IN C#

1. Arithematic Operators:-

(a) Addition operator → +

(b) Subtraction " → −

(c) Multiplication " → *

(d) Division Quotient operator → /

(e) Division Remindu operator → %

2. Comparision Operators:-

(a). Is Equal to → ==

(b). Lessthan → <

(c) Grater than → >

(d). Lessthan Equal → <=

(e). Grateuthan Equal → >=

(f). Not Equal → !=

(g). Data type Comparision → is

3. Logical operators :-

(a) and → &&

(b) OR → ||

(c). Not → !

4. Assignment operators:-

(a). Simple Assignment operator → =

(b). Addition " " → +=

(c) Subtraction " " → −=

(d). Multiplicative Assignment Operator → *=

(e). Division coefficient " " → /=

(f). Division Remindu " " → %=

5. Unary operators :-

(a) Increment → ++

(b). Decrement → --

Explanation of Assignment Operators:-

(a) int a=10; a [10]

a = a+5; a [15]

or

a+ = 5;

(b). int a=10; a [10]

a = a-5; a [5]

or

a- = 5;

(c). int a = 10; a [10]

a = a*5; a [50]

or

a* = 5;

(d). int a = 10; a [10]

a = a/5; a [2]

or

a/ = 5;

(e). int a = 10; a [10]

a=a%5;

or a [0]

a% = 5;

Increment Explanation:-

int a=10; a [10]

a = a+1;

or a [11]

a+= 1;

or

a++ ; → Execution is faster.

Decrement Explanation:-

int a=10 ; a [10]

a=a-1;

or

a-=1;

or

a-- ; → Execution is faste

post Increment

int a=10;

console.Write ("value of a is:- "+(a++));

o/p: value of a is:- 10;

In memory a [11]

pre Increment :-

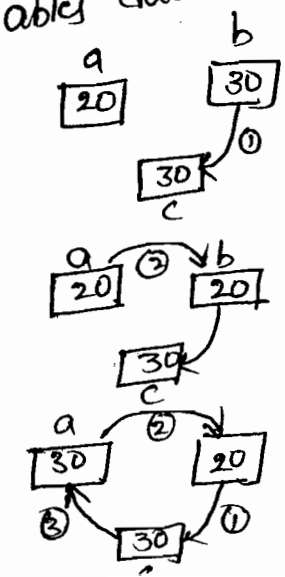
int a=10;

Console Write ("value of a is:- " + (++a));

In memory :- a [11]

o/p: value of a is :- 11

1. Write a program to swap any two variables data



prog:- Method I:-

```
Using System;
Class Swap
{
  Static Void Main( )
  {
    int  a=20; b=30;
    int  c=30;
```

```
Console.WriteLine("Values of a and b are:-"+a+","+b);
        int c = b;
        b = a;
        a = c;
Console.WriteLine("values of a and b after swapping:-"+a+","
                                                    +b);
}
}
```

a	b
20	30

① a = a+b

a	b
50	30

② b = a-b

a	b
50	20

③ a = a-b

a	b
30	20

IInd Method:-

```
Using System;
class Swap2
{
  Static Void Main()
  {
    int a=20, b=30;
    Console.WriteLine("Values of a and b are:-"+a+","+b);
        a = a+b;
        b = a-b;
        a = a-b;
    Console.WriteLine("Values of a and b after swapping:-"+a
                                                    +","+b);
  }
}
```

programming Construct's :—

The methods that we use to write the program in any prog. language are known as programming Constructs.

In any prog. language there are 3 types of prog. Constructs available.

(1) Sequentials
(2) Selections
(3) Iterations.

(1) Sequentials:—

In this method if there are "N" Number of statement's every statement will be Executed one by one without leaving any of the Statements.

so far all the program's we discussed come under Sequential only

(2) Selections:—

In this method in the given set of statement's all the Statement's will n't be executed based on the condition Some set of statement's will be executed and Some other set of statement's will be ignored.

Types of Selections Statement's:-

(a) Simple-If
(b). If-Else
(c), Multiple Ifs
(d) Nested Ifs
(e). Switch Case

(3), Iterations :-

In this method Required number of statements will be Executed repetedly again and again for required number of times.

Types Of Iterative Statements:-

(a), For Loop
(b), While Loop
(c), DO-While Loop
(d), For-Each Loop.

3/10/2014

(a). Simple-If :-

Syntax:-
```
if (condition)
{
  statements;
}
```

Example on Simple-If :-

A person has borrowed some amount of rupees from a lender. Lender said if the amount is returned within 6 months not to pay any interest. Otherwise to pay an interest of

31. for the entire period.

→ Write a program to accept principal, time period calculate intrest amount and total amount is to be paid, based on Simple Intrest.

program:-

```
Using System;
class simple
{
  Static Void Main( )
  {
    double P, T, R=0, I, A;
    Console.Write ("Enter principal and Time period:-");
    P= Convert.ToDouble (Console.ReadLine());
    T= Convert.ToDouble (Console.ReadLine());
    if (T>6)
    {
      R=3;
    }
    I = (P* T*R)/100;
    A = P+I;
    Console.WriteLine ("Intrest is:-"+I);
    Console.WriteLine ("Amount is:-"+A);
  }
}
```

In the above Example, when we compile the program we get an error like "Use of unassigned local variable 'R'."

This is because we can't use a local variable in csharp without initializing it.

For this purpose initialize R is equal to 0 i.e., R=0.

Note:-

```
class C1
{
 int a,b; → Global Variables.
 Void F1( )
 { int p,q; → Local Variables for F1 function.
   if (condition)
   {
     int x, y; → Local Variables for If block
       !
   }
 }
 Void F2( )
 {
   int d,e; → Local Variables for F2 function.
     !
   }
 }
 !
}
```

→ write a program to accept book name, author name and M.R.P of the book from the User, calculate discount amount and selling price of the book. Allow a discount % of 15 incase MRP > 500.

program:-

```
Using System;
Class BOOK
{ Static Void Main()
```

```
{ String    BName, AName
  double    MRP, DP=0, DA, SP;

  Console.WriteLine("Enter the Book name, Author name
                     and M.R.P:-");

  ~~B/=Convert.ToDouble(Console.ReadLine());~~

  BName = Console.ReadLine();

  AName = Console.ReadLine();

  MRP = Convert.ToDouble(Console.ReadLine());

  if (MRP > 500)
  {
    DP = 15;
  }

  DA = (MRP*DP)/100;

  SP = MRP - DA;

  Console.WriteLine("Discount Amount:- "+DA);

  Console.WriteLine(" Selling price :- "+SP);
 }
}
```

.NET FRAMEWORK	VS.NET
1. Execution/Runtime Environment of .NET programs / Applications.	1. Designing / Developing environment of .NET programs/ Applications.
2. Free software	2. Not free software
3. Occupies a memory of 150MB to 500MB.	3. Occupies memory of 1.5GB to 8GB without MSDN is help software and occupies 4GB Memory
4. Versions:-	Versions:-
→ .NET framework 1.0 (Released on feb 13th 2002)	4. → VS.NET 2000 (Released on feb 13th 2002)
→ .NET framework 1.1 (Released on April 3rd 2003)	→ VS.NET 2003 (Released on April 3rd 2003)
→ .NET framework 2.0 (Released on NOV 7th 2005)	→ VS.NET 2005 (Released on NOV 7th 2005)
→ .NET framework 3.0 (Released on NOV 6th 2006)	→ ——— NA ———
→ .NET framework 3.5 (Released on NOV 19th 2007)	→ VS.NET 2008 (Released on NOV 19th 2007)
→ .NET framework 4.0 (Released on April 12th 2010)	→ VS.NET 2010 (Released on April 12th 2010)
→ .NET framework 4.5 (Released on Aug 15th 2012)	→ VS.NET 2012 (Released on Aug 15th 2012)
5. compatable for OS:- Windows 7 SP1 / Windows 8	5. compatable for OS:- Windows 7 SP1 / Windows 8.

Entering into Visual Studio . NET [VS.NET]

→ click on start

→ click on programs

→ click on microsoft visual Studio 2012

→ click on visual Studio 2012.

→ Working with console applications:-

console applications don't provide any GUI facilities.

When we work with console application, we work in CUI (Character User Interface) mode

Creating a New console Application:-

click on file
↓
click on New
↓
click on project
↓
Select visual c# from installed templates.
↓
Select console application template
↓
Type the application name (CA Basics)
↓
choose the location to save
↓
click on OK.

→ Any application will create from visual Studio.Net is treated as a solution.

→ Solution file will have a default extension of ".sln".

→ In visual studio.Net we find so many windows arround, called "Dockable windows."

→ In this dockable windows solution explorer is more useful.

→ Solution Explorer is very much similar to windows explorer. Using ~~windows~~ solution explorer we can perform following tasks.

1. create Files/Folders
2. copy " / "
3. Rename " / "
4. Delete " / "
5. move " / "
6. Browse " / "

→ Go to solution explorer
↓
select program.cs file
↓
clicks with right mouse button
↓
click on rename
↓
change the file name from program.cs to Example1.cs.
↓
write the following code

Ex:-

```
namespace CABasics
{
  class Example1
  {
    Static Void Main()
    {
      Console.WriteLine("welcome");
      Console.ReadLine();
    }
  }
}
```

For first compilation of the program use the following steps.

click on build
↓
click on Build solution } F6

For second compilation at running of the program use the following steps.

click on Debug
↓
Click on start debugging } F5

(2) Working With If-else:-

Syntax:-

```
if (condition)
{ statements;
  :
}
else
{ statements;
  :
}
```

Example:- write a program to find the given number is even or odd?

program:-

```
Namespace CABasics
{
  class Example27
  {
    Static Void Main()
    {
      Console. Write("Enter any number:-");
      int a = Convert.ToInt32 (Console.ReadLine());
      if (a % 2 == 0)
      {
        console. WriteLine("The number is even");
      }
      else
      {
        console. WriteLine(" The number is odd");
      }
      c. ReadLine();
    }
  }
}
```

steps to add a new class file into the application:-

Goto solution Explorer

↓

Select the solution

↓

Click with RM Button (Right mouse)

↓

click on Add

↓

click on New item

↓

select class template

↓

Type the class name [within paranthysis] Example27.cs

↓

↓

Click on Add

Steps to change the Startup object:-

Goto Solution Explorer

↓

Double click on properties

↓

Select application from left side

↓

Goto Startup object

↓

Select the class name to run.

If you don't find class name here. There are 2 possibilities Reasons.

(1), class doesn't contain Main method

(2), Though class contain main method, Main method spelling is incorrect

Ex:- A person Barrowed some amount of rupees from lender. Lender Said if the amount is returned within 6 months, To pay an intrest of 2%, If amount is returned after 6 months to pay an interest of 3%.

write a program to accept principal, time period. calculate Intrest amount, Total Amount and print on the Screen.

→ write a program to find the gratest number between given TWO numbers.

13/04/

13) Multiple-If's:-

Syntax:-

```
if (condition)
{
  statements;
}
if (condition)
{
  statements;
}
```

14) Nested-If's:-

Syntax:-

I Method :-

```
if (condition)
{
  statements;
  if (condition)
  {
    statements;
  }
  else
  {
    statements;
  }
}
else
{ statements;
  if (condition)
  { statements;
  }
  else
  { statements;
  }
```

(OR)

II Method :-

```
if (condition)
{ statements;
}
else if (condition)
{ statements;
}
else if (condition)
{ statements;
}
else
{ statements;
}
```

→ write a program to accept any alphabet and find it is vowel or not?

String → " "
Char → ' '
Number → NO Quotes.

program:-

```
namespace CABasics
{
class Example29
{
Static Void Main()
{
Console.Write("Enter any alphabet:-");
String S= Convert.Console.ReadLine();
if (S=="a")
{ console.Writeline("Vowel");
}
else if (S=="e")
{ console.Writeline("Vowel");
}
else if (S=="i")
{ console.Writeline("Vowel");
}
else if (S=="o")
{ console.Writeline("Vowel");
}
else if (S=="u")
{ Console.WriteLine("Vowel");
}
else
{ console.WriteLine("Not Vowel");
}
consde.Read();
}
}
```

→ write a program to accept any single digit number from the User and print the message in words.

Result:- 0 → zero
1 → One
2 → TWO
3 → Three
4 → FOUR
⋮
9 → Nine
>9 → Not a single digit Number

program:-

```
namespace CABasics.
{ class Example 30
  {
    static Void Main( )
    {
      console. Write("Enter any single digit number:-");
      int a= convert. TOInt32 (Console. ReadLine());
      if (a==0)
      { console. Writeline("zero");
      }
      else if (a==1)
      { console.Writeline("one");
      }
      else if(a==2)
      { console. Writeline("TWO");
      }
      else if (a==3)
      { console. Writeline("Three");
      }
      else if (a==4)
      { console. Writeline ("FOUR");
      }
      else if (a==5)
      { console. Writeline("Five");
      }
```

```
else if (a==6)
{ console.WriteLine("six");
}
else if (a==7)
{ console.WriteLine("Seven");
}
else if (a==8)
{ console.WriteLine("Eight");
}
else if (a==9)
{ Console.WriteLine("Nine");
}
else
{ console.Writeline("Not a single digit Number");
  Console.ReadLine();
}
}
}
```

→ write a program to accept number of sides in a figure and print its name

No. of lines		print
0	→	point
1	→	Single line
2	→	Intersecting / parallel lines
3	→	Triangle
4	→	Quadrilateral
5	→	pentagon
6	→	Hexagon
7	→	Septagon
8	→	octagon

program:-

```
namespace CABASICS
{ class Example31
  {
    Static Void Main()
    {
    Console.Writeline("Enter any number:-");
    int a = Convert.ToInt32(Console.
                Readline());
```

```
if (a==0)
{ console.WriteLine("point");
}
else if (a==1)
  console.WriteLine(" Single Line");
else if (a==2)
   console.Writeline("Intersecting/parallel lines");
else if (a==3)
   Console.WriteLine(" Triangle");
else if (a==4)
   console.WriteLine(" Quadrilateral");
else if (a==5)
   Console.WriteLine(" pentagon");
else if (a==6)
   Console.WriteLine(" Hexagon");
else if (a==7)
   Console.WriteLine("Septagon");
else if (a==8)
   Console.WriteLine("octagon");
else if (a==9)
   Console.Writeline("Nanogon);
else if (a==10)
   Console.Writeline("Decagon");
else
  {console.WriteLine ("polygon with more than 10 sides");
  }Console.ReadLine();
}
}
```

9 → Nanogon

10 → Decagon

>10 → polygon with more than 10 sides.

→ write a program to accept age of a person and print the appropriate the message like

if age upto 2 years → you are an infant
age > 2 to 5 years → you are a kid
age > 5 to 12 years → you are a child.
age > 13 to 19 years → you are a tenager
age > 19 to 35 years → you are a young star
age > 35 to 50 years → you are a middle age
age > 50 years → you are in old age

program:-

```
namespace CABASICS
{
Class Example32
{
  Static Void Main( )
  {
    Console. Writeline ("Enter the age of a person:-");
    Age a = Convert.ToInt32 (Console.ReadLine());
    if (age <= 2)
    { console. Writeline ("you are an infant");
    }
else if (age <= 5)
    { Console. Writeline ("you are a kid");
    }
else if (age <=12)
    { console Writeline ("you are a child");
    }
```

```
else if (age <= 19)
  { console.Writeline ("you are a teenager");
  }
else if (age <= 35
   Console. WriteLine("you are a youngstar");
else if (age <= 50)
   Console. Writeline ("you are a middle age");
else
   Console. WriteLine(" you are in old age");
   Console. ReadLine();
  }
 }
}
```

→ write a program to accept any number upto 6 digits and print number of digits in that without using loops.

→ write a program to accept customer name, meter number, previous reading, present reading for electricity department. calculate no. of units and Bill amount is to be paid. use the following conditions.

(i), For units upto 100 Rs 1 per unit
For units " 200 Rs 2 " "
" " " 300 Rs 3 " "
" " " 400 Rs 4 " "
" " " 500 Rs 5 " "
" " > 500 " 6 " "

Ex: for (i)

Assume units 65 → 65 * 1 → 65

" 130 → 130 * 2 = 260

" 240 → 240 * 3 = 720

" 320 → 320 * 4 = 1280

" 450 → 450 * 5 = 2250

" 560 → 560 * 6 = 3360

FOR (ii) Units 0 to 100 Rs 1 per unit

" 101 to 200 Rs 2 " "

" 201 to 300 Rs 3 " "

" 301 to 400 Rs 4 " "

" 401 to 500 Rs 5 " "

" > 500 Rs 6 " "

Units → 65 → 65*1 → 65

" → 130 → (100*1) + (130 − 100) * 2 → 100 + 60 → 160

" → 240 → (100*1) + (100*2) + (240 − 200) * 3 → 100 + 200 + 120 → 420

" → 320 → (100*1) + (100*2) + (100*3) + (320 − 300) * 4 → 100 + 200 + 300 + 80 → 680

" → 450 → (100*1) + (100*2) + (100*3) + (100*4) + (450 − 400) * 5 → 100 + 200 + 300 + 400 + 250 → 1250

" → 560 → (100*1) + (100*2) + (100*3) + (100*4) + (100*5) + (560 − 500) * 6 → 100 + 200 + 300 + 400 + 500 + 360 = 1860.

program for Ex(i):-

```
namespace CABASICS
{class Electricity
 {
  Static Void Main( )
  {
```

program for Ex(ii):-

16/10/17

```
namespace CABASICS
{ class Example34
  {
    Static Void Main()
    {
      String CName, MNum; int Prs Read, Prs Read;
      int units; double Bill;
      Console.Write("Enter CName, MNUM, Prv Read and
                                        Prs Reading :-");
      CName = Console.ReadLine();
      MNUM = Console.ReadLine();
      Prv Read = Convert.To Int32 (Console.ReadLine());
      Prs Read = Convert.To Int32 (Console.ReadLine());
      Units = Prs Read - Prv Read;
      if (units <=100)
          Bill = units*1;
      else if (units <= 200)
          Bill = 100+(units-100)*2;
      else if (units <=300)
          Bill = 300+(units-200)*3;
      else if (units <=400)
          Bill = 600+(units-300)*4;
      else if (units <=500)
          Bill = 1000+(units-400)*5;
      else
          Bill = 1500+(units-500)*6;
```

```
        Console.WriteLine("No. of units are :- " + units);
        Console.WriteLine(" Bill Amount is :- " + Bill);
    }
}
}
```

Working with Logical operators:-

(1). AND (&&) operator:-

In the given set of statements if every statement is returning true then and operator will return True. If any single statement is returning false and operator will return false.

Ex:- and &&

Statement 1: Goto Theatere

Statement 2: purchase Tickets

Result : watch the movie

Stmt 1	Stmt 2	Result
T	F	F
F	T	F
F	F	F
T	T	T

(2) OR operator:-

In the given set of statements, if any single statements return's true OR operator will return true. If all the statements returning false then OR operator return false.

OR ||

Ex:-

Stmt 1 : Goto Theatere By bike

Stmt 2 : Goto Theatere By car

Result : watch the movie

Stmt1	Stmt2	Result
T	F	T
F	T	T
T	T	T
F	F	F

(11). write a program to find the given alphabet is vowel or not?

program:-

```
namespace CABASICS
{
class Example35
{
Static Void Main()
{
Console.Write("Enter any alphabet:");
String S= Console.ReadLine();
if (S=="a"||S=="e"||S=="i"||S=="o"||S="u")
    Console WriteLine ("Vowel")
else
    Console Writeline ("Not a Vowel")
    Console. Read();
}
}
}
```

S == "a"
S == "e"
S == "i"
S == "o"
S == "u"

[any single stmt is True. Result is True so use OR operator]

12) write a program to accept Student roll no, name and marks in 4 Subjects. Find total marks and result and print on the Screen to User. ~~In all~~

In all Subjects if user get >=35, then result is pass otherwise result is fail.

Condition is
S1 >= 35
S2 >= 35
S3 >= 35
S4 >= 35

program:-
```
namespace CABASICS
{
class Example36
{
Static Void Main( )
{
String SName; Result;
int RNO, S1, S2, S3, S4; Total;
Console. WriteLine("Enter SName, RNO, marks in S1, S2,
                    S3 and S4:-));
SName = Console. Readline( );
RNO = Convert. ToInt32 (console. Readline());
S1 = Convert. ToInt32 ( "      " );
S2 = Convert. ToInt32 ( "      " );
S3 = "  "  "  ( "      " );
S4 = "  "  "  ( "      " );
Total = S1+S2+S3+S4;
if (S1>=35 && S2>=35 && S3>=35 && S4>=35)
Result = pass;
else
Result = fail;
C. Write Line ("Total Marks are:-" +Total);
```

```
        Console.WriteLine("Result is:-" + Result);
        Console.Read ();
    }
  }
}
```

H.W → write the above program by using OR operator Find total marks and result.

→ write a program to accept name, age and gender of a person, and print the person is major or minor

conditions:- Name
age
Gender → M|F

Age >= 18 | Major
G → F

Age >= 21 | minor.
G → M

program:-
```
namespace CABASICS
{
    Class Example37
    {
        Static Void Main()
        {
            String name, gender;
            int age;
```

(e). Working With Switch Case:-

Syntax:-

```
switch (Expression)
{
    case Expr1 :
        Statements;
        :
        break ;
    case Expr2 :
        Statements;
        :
        break
    :
    default :
        statements;
        :
    }   break;
```

Note1: In the Expression we supply the value what is to be compared. In Every case we supply the value with which comparision should be made

The Expression in switch will compare with the value avialable in every case.

With which case value the switch Expression matches then that case will be executed, remaing all cases, default will be ignored.

Note 2:- If switch Expression value doesn't match with any of the case Expression values. Then all cases will be ignored and default will be Executed

Limitations of switch Case:-

(1). Switch case can be used only with equal to comparision.

(2). Logical operators can't be used with switch case

Examples with Switch Case:-

Hot 4

(1) write a program to find the given alphabet is vowel or not?

program:-
```
namespace CABASICS
{
class Example38
{
Static Void Main()
{
```

```
Console. WriteLine ("Enter any alphabet");
String S = Console.ReadLine();
Switch(S)
{ case "a":
      C.WL ("Vowel");
      break;
  case "e":
      C.W.L ("Vowel");
      break;
  case "i":
      C.W.L ("Vowel");
      break;
  case "o":
      C.W.L ("Vowel");
      break;
  case "u"
      C.W.L ("Vowel");
      break;
  default:
      C.WL ("Not a Vowel");
      break;
  }
  Console. Read();
 }
}
}
```

In the above Example, Console.Writeline("Vowe"); and break these 2 statements are repeated for 5 times. To overcome this drawback we write the code like

```
namespace CABASICS
{
  class Example38
  {
    Static Void Main()
    {
      Console.Write("Enter any alphabet:-");
      String S = Console.ReadLine();
      Switch(S)
      {
        case "a";
        case "e":
        case "i":
        case "o":
        case "u":
         C.WL("Vowel")
         break;
        default:
         C.WL("Not a Vowel");
        break;
      Console.Read();
    }
  }
}
```

(2) Write a program to accept any single digit number and print it in words.

```
namespace CABASICS
{ class Example 39
  {
    Static Void Main()
    { console.Writeline("Enter any Single digit Number:-");
```

```
String S = Console.ReadLine();
Switch (S)
{
    case "1":
    Console.WriteLine ("one");
            break;
    case "2":
    Console.WriteLine ("TWO");
            break;
    Case "3":
    Console.WriteLine ("Three");
            break;
    Case "4":
    Console.Writeline("FOUR");
            break;
    case 5:
    Console.Writeline("Five");
            break;
    case 6:
    Console.WriteLine("Six");
            break;
    case 7:
    Console.WriteLine("Seven");
            break;
    case 8:
    Console.Writeline("Eight");
            break;
    case 9:
    Console.WriteLine("Nine");
    break;
    Default:
    Console.Writeline("Not a single digit Number :-");
    break;
    }
    Console.Read();
  }
}
}
```

(3), Write a program to accept any two numbers from the user and perform the arithmatic operations according to User choice.

program:-

```
namespace CABASICS
{
 class Example40
 {
  Static Void Main()
  {
   int a, b. Result
   Console.Writeline("Enter any two numbers ~~& your choice~~ :-");
   a = ~~To~~ Convert.ToInt32(Console.Readline());
   b = Convert.ToInt32(Console.Readline());
   ~~Result = Convert.ToInt32 (Console.Read(line()));~~
   ~~switch (choice)~~ C.WL("1 Add\n 2 Subtract\n 3 multiply\n 4 Divide");
   ~~case~~
    C.Write("Enter your choice:-");
    int opt = Convert.ToInt32(C.RL());
    Switch (opt)
    { case 1:
      C.WL("sum is:-" +(a+b));
      break;
      case 2:
      C.WL("Diffuence is:-" +(a-b));
      break;
```

Enter any two numbers:-
90 10
1. Add
2 Subtract
3. multiply
4. Divide
Enter your choice:-3
product is:- 900

```
        case 3:
            C.WL("product is:- "+(a*b));
            break;
        case 4:
            C.WL("Quotient is :-" + (a/b));
            break;
        default:
            C.WL("Invalid option");
            break;
        }
        Console.Read();
    }
}
}
```

Working with Loops:-

(a) For Loop:-

Syntax:-

```
for(variable = Start Value; condition; Increment/Decrement)
{
    Statements;
    :
}
```

Step Execution of for Loop:-

Step1: variable = Start value will be Executed.

Step2: Condition will be checked.

Step3: If condition is true, enters into the loop executes all the statements

Step4: When closing delimeter is encountered, control shifts to Increment/decrement and variable will be Incremented/Decremented.

Step 5: Again condition will be checked. As long as condition is true loop will be repeated. When condition is false loop will be terminated.

(1), Example to print "welcome" on the Screen to User.

output:-

1 Welcome
2 Welcome
3 Welcome
4 Welcome
5 Welcome

```
namespace CABasics
{ class Example41
  {
    Static Void Main()
    {for(int i=1; i<=5; i=i+1)
      {
        Console.WriteLine("welcome");
      }
      Console.Read();
    }
  }
}
```

(2) Write a program to print first 5 Natural Number on the Screen to User.

output:- 1 2 3 4 5

```
namespace CABasics
{ class Example42
  {
    Static Void Main()
    {for(int i=1; i<=5; i=i+1)
      {
        Console.Write(i + " ");
      }
      Console.Read();
    }
  }
}
```

→ Console.WriteLine("First 5 Natural Numbers are:");

or i++

13). Write a program to print first 5 Natural Numbers in reverse order

output:- 5 4 3 2 1

```
namespace CABasics
{
  class Example
  {
    Static Void Main()
    {
      for(int i=5; i>0 (or) i>=1, i--)
      {
        Console.Write(i + " ");
      }
      Console.Read();
    }
  }
}
```

i	i>=1
5	5>=1
4	4>=1
3	3>=1
2	2>=1
1	1>=1

5 4 3 2 1

18/10/2014

14). Write a program to print following output like 0 2 4 6 8

program:-

```
namespace CABasics
{ class Example
  {Static Void Main()
   {
    for(int i=0; i<=8; i=i+2)
    {Console.Write(i + " ");
    }
    Console.Read();
   }
}
```

i	i<=8
0	2<=8
2	4<=8
4	6<=8
6	

0 2 4 6 8

(5) Write a program to print first five multiples of 3"

i	i<=15
3	3<=15
6	6<=15
9, 12, 15	9<=15

i= 3 6 9 12 15

program:-

```
namespace CABasics
{
  Class Example
  {
    Static Void Main()
    {
      for (int i=3; i<=15; i=i+3)
      {
        Console.Write(i + " ");
      }
      C.Read();
    }
  }
}
```

(OR)

```
for(int i=1; i<=5; i++)
{
  Console.Write((3*i) + " ");
}
```

output:- 3 6 9 12 15

3 3*2 3*3 3*4 3*5

output:- 3 6 9 12 15

Note:- The above method is n't suitable for different values multiples, so developers we use generalized method.

Generalized Method:-

```
namespace CABasics
{ class Example
  {
    Static Void Main()
    {
      Console.Write ("Enter any Number:-");
      int Num = Convert.ToInt32 (Console.ReadLine());
      Console.WriteLine ("First 5 multiples of "+Num+" are:-");
      for(int i=1; i<=5; i++)
      {
        Console.Write((Num*i) + " ");
      }
      Console.Read();
    }
  }
}
```

16). Write a program to print multiplication table of the given number.

program:- namespace CABasics

```
{
 class Example49
 {
  Static Void Main()
  {  int Result;
     Console.Write(" Enter any Number:-");
     int Num= Convert.ToInt32 (Console.ReadLine());
     for( int i=1 ; i<=10 ; i++)
     {
       Console.Write(Num+"*"+ i +"=" + (Num*i));
       }
     Console.Read();
  }
 }
}
```

9

1<=10
2<=10

9*1 = 9
9*2 = 18
9*3 = 27
9*4 = 36
9*5 = 35
⋮
9*10 = 90

Num*i = Num*;

In output Stream arguments method:-

```
for(int i=1; i<=10; i++)
{
 console.write("{0} * {1} = {2}", Num, i, (Num*i));
}
```

(7) write a program to print following structure.

Even	odd
0	1
2	3
4	5
6	7
8	9
10	

i	N	E	O
i=0	10	0	1
~~0~~ → 0<=10		2	3
~~1~~ → 1<=10		4	5
~~2~~ → 2<=10		6	7
~~3~~ → 3<=10		8	9
~~4~~ → 4<=10		10	
~~5~~ → 5<=10			
~~6~~ → 6<=10			
~~7~~ → 7<=10			
~~8~~ → 8<=10			
9 → 9<=10			
10 → 10<=10			

program:- namespace CABasics

```
{ class ExampleSD
  {
   Static Void Main()
   {
    Console.WriteLine("Even\tOdd\n----------");
    for(int i=0; i<=N; i++)
    {
     if(i%2==0)
      Console.Write(i + '\t');
     else
      Console.WriteLine(i);
    }
    Console.Read();
   }
  }
}
```

(8) write a program to find sum of the first "N" Natural Numbers.

Ex:- User enter N=5

1+2+3+4+5 = 15

sum of the first 5 Natural Number is :- 15

20/10/14

code:- namespace CABasics

```
{
 class Example59
 {
  Static Void Main()
  {
   int N, Sum=0;
   Console.Write (" Enter N value :- ");
   N = Convert.ToInt32 (Console.ReadLine());
  for (int i=1 ; i<=N; i++)
   {
    Sum = sum+i ; (OR) sum+=i;
   }
   Console.WriteLine("sum of first "+N+ Natural Numbers is:- +sum);
   Console.Read();
  }
 }
}
```

N	SUM	i
5	~~0~~	~~1~~
	~~1~~	~~2~~
	~~3~~	~~3~~
	~~6~~	~~4~~
	~~10~~	~~5~~
	15	

output:- sum of first 5 Natural Numbers is:- 15.

(9). write a program to find sum of the squares of the first 'N' Natural Numbers.

i.e., $1^2+2^2+3^2+4^2+5^2=55$

code:- namespace CABasics

```
{
 class Example52
 {
  Static Void Main()
  {
   int N,
```

(10). write a program to find the factorial of the given number.

code:-
```
namespace CABasics
{
  class Example53
  {
    static Void Main()
    {
      int N, Fact = 1;
      Console. WriteLine (" Enter N value:");
      N = Convert. To Int32 (Console. ReadLine());
      for (int i=1; i<=N; i++)
      {
        fact = fact*i;
      }
      Console. WriteLine (" Factorial Of"+N+"is:");
      Console. Read ();
    }
```

11), write a program to find factors of the given Number.

i.e., output:- 1) 8 (8 2) 8 (4 3) 8 (2 4) 8 (2 5) 8 (1 6) 8 (1

8 8 6 8 5 6

0 0 2 0 3 2

7) 8 (1 8) 8 (1

7 8

1 1

code:-
```
namespace CABasics
{
  class Example54
  {
    Static Void Main()
    {
      Console.Write("Enter Any Number:-");
      int N = Convert.ToInt32(Console.ReadLine());
      Console.Write("Factors of" + N "+are:-");
      for (int i = 1; i<= N; i++)
      {
        if (N%i ==0)
        {
          Console.Write(i + "  ");
        }
      }
      Console.Read();
    }
  }
}
```

output:- factors of 8 are:- 1 2 4 8.

12), write a program to find sum of the squares of factors of a given number.

i.e, $1^2+2^2+4^2+8^2 = 85$.

(13). Write a program to find the given number is perfect Number or Not?

Perfect Number:- perfect Number is, if sum of the factors is double to the number, then it is called perfect Number.

(14). write a program, count of the factors of the given number.

i.e, count of the factors of 8 is :- 4

(15). write a program, the given number is prime number or not?

prime number means, a number has two factors, those are 1 and itself.

(14). namespace CABasics

```
{ class Example57
  { static void Main()
    {
      int N;
      Console.Write("Enter Any Number:-");
      int N = Convert.ToInt32(Console.ReadLine());
      Console.Write("Factors Of " +N "are":- ");
      for (int i=1; i<=N; i++)
      {
        if (N%i==0)
```

(16) Write a program to find sum of the digits in a given number.

```
code:- namespace CABasics
{
  class Example59
  {
    Static Void Main()
    {
      int Num, SUM=0
      Console.Write("Enter Any Number:-");
      Num Convert.ToInt32(Console.ReadLine());
      for( int N=Num; N>0; N=N/10):
      {
        SUM = SUM+(i)+i;
        Console.Write(
        Num = N/10;
        Sum = Sum+ N/10;
      }
      Console.Write("sum of the digits of" +N"+ is :-"
                                                Sum);
      }
      Console.Read();
    }
  }
}
```

Sum = 0
5
5+2=7
7+6=13
13+3=16

10) 3625 (362
3620
5

10) 362 (36
360
2

10) 36 (3
30
6

10) 3 (0
0
3

NUM	SUM	N
3625	0	3625
	5	362
	7	36
	13	

output:- Sum of the digits of 3625 is :- 3+6+2+5 = 16.

(17) Write a program to count the digits of a given Number

```
code:- namespace CABasics
{
  class Example60
  {
    Static Void Main
    {
      int Num, count=0;
```

```
Console.Write("Enter Any Number:-");
int N = Convert.ToInt32(Console.ReadLine());
Count = Convert.ToInt32(Console.ReadLine());

for(int N=Num; N>0; N=N/10)
  {
    count++;
  }
Console.Write("Number has" + count + "Digits":-);
Console.Read();
  }
}
}
```

Num	count	N
3625	~~3~~	3625
	6	362

output:- Num of the digits in 3625 is :- 4

(18) Write a program to find reverse Number of the given number

```
Code:- namespace CABasics { class Example61
  {
    Static Void Main()
    {
      int RNum, RNum=0;
      Console.Write("Enter Any Number:-");
      Num = Convert.ToInt32(Console.ReadLin())
      for(int N=Num; N>0; N=N/10)
        {
          ~~N /= Num%10 * 10~~;
          ~~RNum= RNum * 10, +N~~;
          RNum = (RNum*10) + N%10;
        }
```

NUM	RNUM	N/10
3625	~~0~~	5
	5	
	56	
	~~5626~~	
	~~5263~~	

RNUM = 0
5
5×10+2=52
52×10+6=526
526×10+3=5263

```
Console.WriteLine("Reverse Number is :-"
                  +NUm);
    Console.Read();
  }
 }
}
```

RNum	Reminder
(0*10)	+ 5
(5*10)	+ 2
(52*10)	+ 6
(526*10)	+ 3 = 5263

19). write a program to find the given number is palindrome or Not?

palindrom:- Given Number is equal = Reverse Number.

Code:-
```
namespace CABasics
{ class Example 62
  { Static Void Main()
    { int NUm,
      Console. Write("Enter Any Number:-");
      int NUm= Convert.ToInt32 (Console.ReadLine());
      for(int N=NUm; N>0; N/10)
      {
        RNUm =(RNUm*10)+N%10;
        if (RNUm = NUm)
        { console. write(" Number is palintromes");
        }
        else
          Console write (" Number is n't palindrome");
      }
      console. Read();
  } } }
```

(20). Write a program to print the Fibonacci series.

i.e, 0 1 1 2 3 5 8 13, 21, etc...

(2). Working With While Loop:-

Syntax:-

```
while (condition)
{
  Statements;
    ⋮
  [Increment|Decrement];→optional
}
```

(1). Write a program to print first 5 Natural Numbers on the Screen to user. i.e, 1 2 3 4 5

Code:-

```
namespace CABasics
{
  class Example64
  {
    Static Void Main()
    {
      int i=1
      while(i<=5)
      Console.Write(i+" ")
      i++;
    }
    Console.Read();
  }
}
}
```

i 1 2 3 4 5
~~1~~
~~2~~
~~3~~
~~4~~
5

output:- 1 2 3 4 5.

22/10/20

(2) write a program to find the given number is even or odd? and repeat the program according to user choice

code:-
```
namespace CABasics
{
 class Example66
 {
  Static Void Main()
  {
   String choice="y";
   while (choice=="y")
   {
    Console. write("Enter Any Number:");
    int N= Convert. ToInt32(Console.ReadLine());
    if (N%2==0)
    Console. WriteLine("The number is even");
    else
     Console. writeline("The number is Odd");
    Console. Write ("Enter y to continue:-");
    choice = Console. ReadLine();
   }
   Console. Read();
  }
 }
}
```

(3) write a program to find the greatest number between two number and repeat the program according to user choice

code:-
```
namespace CABasics
{ class Example67.
```

```
Static Void Main()
{
    String choice = "y";
    while (choice == "y")
    {
        console.writeline("Enter Any Two Numbers:-");
int Num1 = Convert.ToInt32(Console.Read Line());
int Num2 = Convert.ToInt32(Console.ReadLine());
    if (Num1 > Num2)
    {
        Console.Write( Num1 + "  ");
    }
    else
        console.write(Num2 + "  ");
```

13 write a program to find the given number is prime number or not? according to user choice?

code:-

```
namespace CABasics
{ class Example68
  {
    Static Void Main()
    { String choice = "y";
```

```
while (choice == "y")
{   int N, count=0;
    Console. Write ("Enter Any Number:-");

    int N = Convert. ToInt32(Conside. ReadLine());

      for(int i=1 ; i<=N; i++)

      {
        if(N%i==0)
          count++;
      }
      if (count == 2)
      Console. WriteLine ("The number is prime");
      else
       Console. WriteLine ("The Number is not prime");
      Console. Write("Enter y to continue:-");

      choice = Console. Read();

    }
    Console Read();
  }
 }
}
```

N	i	count
5	1	1
	2✗	
	3✗	
	4✗	
	5✓	

(A) Write a program to find the factorial value of the given number and repeat the program according to user choice.

Code:-

```
namespace CABasics
{
class Example69
{
Static Void Main()
{
    String choice="y";
    while (choice=="y")
    {
      int N, Fact=1;
      Console.Write("Enter Any Number:-");
      N= Convert.ToInt32(Console.~~Write~~ReadLine());
      for(int i=1; i<=N; i++)
      {
        fact= fact*i;
      }
      Console.WriteLine("Factorial of "N+"is" +fact);
      Console.Write("Enter y to continue:-");
    } choice =c.Read();
    Console.Read();
  }
}
}
```

Working with Do-while Loop :-

Syntax:-
```
do
{
  Statements;
  !
  [Increment/Decrement]
} while (condition);
```

do-while loop executes minimum ones without checking the condition. At the end of the loop condition will be checked, as long as loop condition is true loop will be repeated when condition is false loop will be terminated.

(1) write a program to find the given number is even or odd according to user choice.

code:-
```
namespace CABasics
{
  class Example10
  {
    Static Void Main()
    {
      String choice;
      do
      {
        Console.write ("Enter Any Number:-");
        int N= Convert.ToInt32 (Console.Readline());
        if(N%2==0)
        Console.writeline("The Number is even");
```

```
else
    Console.WriteLine("The Number is Odd");
    Console.Write ("Enter y to Continue:-");
    choice. Console.ReadLine();
  } while (choice == "y");
  Console.Read();
  }
}
}
```

27/10/2014

Working with Nested Loops:-

(1) write a program to print the following structure

```
*
* *
* * *
* * * *
* * * * *
```

	1	2	3	4	5
1	*				
2	*	*			
3	*	*	*		
4	*	*	*	*	
5	*	*	*	*	*

R		C
1	1s	1
2	2s	1 2
3	3s	1 2 3
4	4s	1 2 3 4
5	5s	1 2 3 4 5

code:-

```
namespace CABasics
{
  class Example-11
  {
    Static Void Main()
    {
      for (int R=1; R<=5; R++)
      {
        for (int c=1, c<=R; c++)
        {
          Console.write("*");
        }
        Console.WriteLine();
      }
      Console.Read();
    }
```

R	C
1	~~1~~ 2
2	~~1~~ ~~2~~ ~~3~~
3	~~1~~ ~~2~~ ~~3~~ ~~4~~
4	~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~
5	~~1~~ ~~2~~

```
*
→ * *
→ * * *
→ * * * *
→ * * * * *
→
```

(2), write a program to print following Structure

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

code:- namespace CABasics

```
{
  Class Example72
  {
    Static Void Main()
    {  int  ki;  → convert.ToInt32(Console.ReadLine());
      for(int R=1; R<=5; R++)
      {
        for(int C=1; C<=R; C++)
        {
          Console.write(" ")
        }
        Console.writeline();
      }
      Console.Read();
    }
  }
}
```

R	C
1	1
	2

(3), write a program to print the following

```
1
2 3
4 5 6
7 8 9 10.
```

	1	2	3	4
1	1			
2	2	3		
3	4	5	6	
4	7	8	9	10

code:- namespace CABasics

R	C	1
1	1	

```
{
  Class Example73
  {
    Static Void Main()
    {
      for(int R=1; R<=10; R++)
      {
        for(int C=1; C<=R; C++)
```

```
Console.Write("Enter Number of rows:-")
int N= Convert.ToInt32(Console.RL())
int value=1;
for(int R=1; R<=N; R++)
{
 for(int c=1; c<=R; c++)
  {Console.Write(Value++);
  }
 }
 Console.WriteLine();
 }
 Console.Read();
 }
}
```

N	value	R	C
4	~~1~~	~~1~~	~~1~~
	~~2~~		2
	~~3~~	~~2~~	~~1~~
	~~4~~		~~2~~
			~~3~~
	~~5~~	~~3~~	~~1~~
	~~6~~		~~2~~
	~~7~~		~~3~~
			~~4~~
	~~8~~	~~4~~	~~1~~
	~~9~~		~~2~~
	10		~~3~~
			~~4~~
			~~5~~

14). write a program to print the multiples of 5 like

```
5
5   10
5   10   15
5   10   15   20
```

code:-

```
console.write("Enter Number of rows:-");
int N= Convert.ToInt32(Console.Read Lin());
int value=5;
for (int R=1; R<=N; R++)
{
 for(int c=1; c<=R; c++)
  {Console.write(value+=value*5+" ");
```

N	Val	R	C
4	5	1	1
			5

15)

```
namespace CABasics
{ class Example#5
  {Static void Main()
   {console.write("Enter Any Number");
    int Num = Convert.ToInt32 (C.RLL());
    Console.write ("Enter Number of rows:-");
    int N = Convert.ToInt32(C.RLL());
    for(int R=1; R<=N; R++)
    {for(int C=1; C<=R; C++)
     {Console.write( C*Num+" ");
     } C.WL();
    }
    Console.Read();
   }
  }
}
```

16) write a program to print following structure.

```
1
2  4
3  6  9
4  8  12  16
```

code:- namespace CABasics

```
{ class Example76
  {
  Static Void Main()
  { C.wr("Enter Number of rows:-");
    int N= Convert.ToInt32 (Console.ReadLine());

    for (int R=1; R<=N; R++)
      {
      for (int C=1; C<=R; C++)
        {
        Console.Write(C*R+" ");
        }
        Console.WriteLine();
      }
      Console.Read();
    }
  }
}
```

	1	2	3	4
1	1			
2	2	4		
3	3	6	9	
4	4	8	12	16

77).

```
2
3 4
4 5 6
5 6 7 8.
```

code:- namespace CABasics

```
{ class Example77
  {
  Static Void Main()
  { C.W("Enter Number of rows:");
    int N= C.ToInt32 (C.RL());
```

```
for(int R=1; R<=N; R++)
    {
      for(int C=1; C<=R; C++
        {
          console.write("R+C+" ");
        }
        Console.writeLine();
      }
      Console.Read();
    }
  }
}
```

(8) write a program to print the following

```
* * * * *
* * * *
* * *
* *
*
```

	1	2	3	4	5
1	*	*	*	*	*
2	*	*	*	*	
3	*	*	*		
4	*	*			
5	*				

R C

code:- namespace CABasics

```
{class Example78
 {static void Main()
  {
   for(int R=1; R<=Lines; R++)
    {
     for(int C=R; C<=Lines; C++)
      {console.write("*");
      }
      C.WL();
    }C.Read()
  }
}
```

→ console.write("Enter Number of lines:");
int Lines = Convert.ToInt32(Console.RL());

N	R	C
5	1	1

(9) Write a program to print the following like

5 4 3 2 1

(10) WAP to print the following

```
        *
      * *
    * * *
  * * * *
* * * * *
```

R		C
1	1S	5
2	2S	4 5
3	3S	3 4 5
4	4S	2 3 4 5
5	5S	1 2 3 4 5

	1	2	3	4	5
1					*
2				*	*
3			*	*	*
4		*	*	*	*
5	*	*	*	*	*

code:-

```
namespace CABasics
{
  class Example 79
  {
    Static Void Main()
      → C.W("Enter Number of lines);
      int Lines = C.ToInt32(C.RL());
      for(int R=1; R<=Lines; R++
      {
        ~~for(int C=5; C<=5; C--)~~
        {
          Console.write("*");
        }
        Console.writeline();
      }
      Console.Read();
    }
  }
}
```

→ for(int space=R; space<Lines; space++)
{ C.write(" ");
}
for(int star=1; star<=R; star++)

R	spaces	stars
1	4	1
2	3	2
3	2	3
4	1	4
5	0	5

R	spaces	stars
1	4	1
2		

(ii), WAP To print the following

```
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
```

	1	2	3	4	5
1					1
2				1	2
3			1	2	3
4		1	2	3	4
5	1	2	3	4	5

code:-

```
namespace CABasics
{
class Example80
{
Static void Main()
{
Console.write ("Enter Number of lines:-");
int Lines = Convert.ToInt32 (Console.ReadLine());
for(int R=1; R<=Lines; R++)
{for(int space = R; space<Lines; space++)
{ c.write(" ");
}
for (int c = 1; c<=R; c++)
{ C.write ("c");
}
} c.WL();
Console.Read();
} } }
```

112) WAP to print the following

```
        1
      2 1
    3 2 1
  4 3 2 1
5 4 3 2 1
```

	1	2	3	4	5
1					1
2				2	1
3			3	2	1
4		4	3	2	1
5	5	4	3	2	1

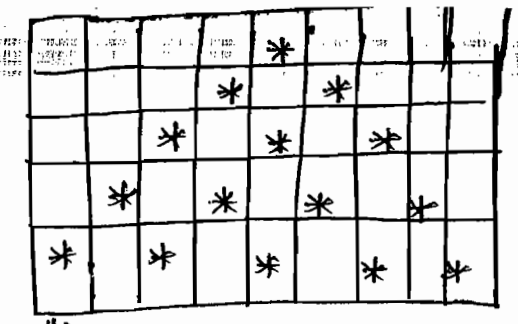
code:-
```
namespace CABasics
{
class Example81
{
Static void Main()
{
console write ("Enter Number of Lines");
int Lines = Convert.ToInt32 (Console.ReadLine());
```

113) WAP To print the following structure

```
    *
   * *
  * * *
 * * * *
* * * * *
```

code:-

```
namespace CABasics
{
class Example82
{
Static Void Main()
{
Console.write ("Enter Number of Lines:");
int Lines = Convert.ToInt32 (C.RL());
for (int R = 1; R<=Lines; R++)
{
for(int space=R; space<Lines; space++)
{ Console.write(" ");
}
for(int star=1; star<=R; star++)
{
Console.write ("*  ");
}
Console.writeLine();
}
Console.ReadLine();
}
}
}
```



R	spaces	star
1	4	1
2	3	2
3	2	3
4	1	4
5	0	5

(14). WAP to print the following structure

```
        1
      1   2
    1   2   3
  1   2   3   4
1   2   3   4   5
```

(15). WAP to print the following structure

```
*                   *
* *                 * *
* * *               * * *
* * * *             * * * *
* * * * *           * * * * *
```

16) WAP to print the following structure

```
        *
      *   *
    *   *   *
  *   *   *   *
*   *   *   *   *
  *   *   *   *
    *   *   *
      *   *
        *
```

17) print the following

```
* * * * *
* * * *
* * *
* *
*
```

18) WAP to print the following structure

```
* * * * * * * * *
* * * *   * * * *
* * *       * * *
* *         * *
*             *
```

19) WAP to print pascal triangle

```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
1   5   10  10  5   1
```

ARRAYS

Working with Arrays:-

An Array is used to store more than one values, with same Name. Each value of the array is identified by using its index value

In general, first index is always begin with "0" and is known as "Lower bound of the Array". Last index of the array will end with "size - 1" of the array, and is known as upper bound of the Array.

→ In csharp array's are reference types.

→ Array's are in general user defined datatypes.

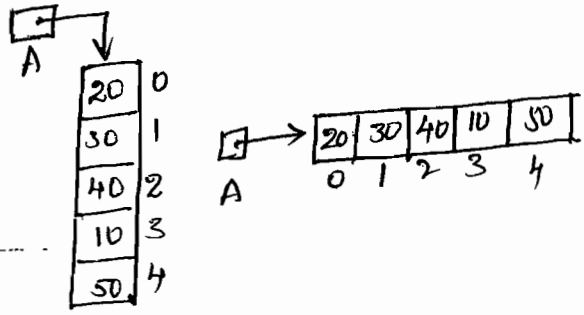
Types of Array's:-

- Types of array's
  - According to physical size
    - single dimensional array's
    - Multi dimensional array's
  - According to memory size
    - Normal Array's
    - Huge array's
  - According to Nature of physical size
    - static array's
    - Dynamic array's
  - special type → Jagged array's

2) Single Dimensional array:

An array that contains, either a single or single column is known as Single dimensional array

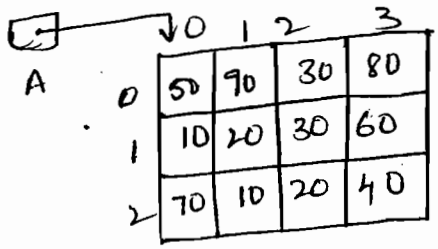
Ex:-



Multi dimensional array:-

An array that contains, more than one rows and more than one column's is known as multi-dimensional array.

Ex:- Example of 2D Array.



Normal Array's:-

An array which occupies memory size of <=64KB is known as normal Array.

Huge Array:-

An array which occupies a memory size of >64KB is known as huge array.

Static Array:-

An array whose size is fixed through-out the program, doesn't changed during runtime is known as static Array.

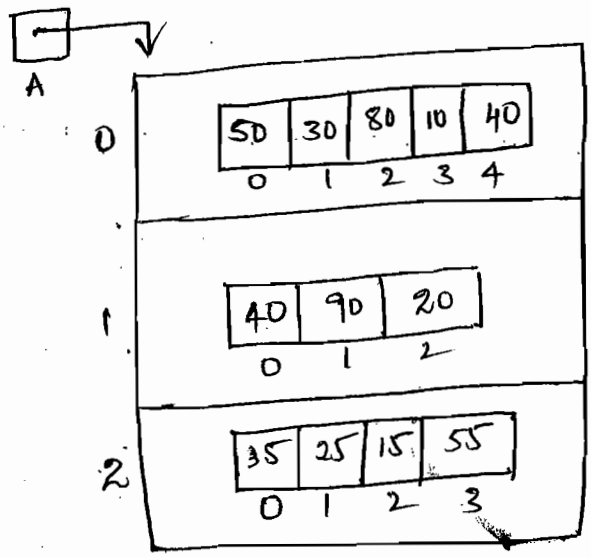
Dynamic array:-

An array whose size is n't fixed throughout the program, and will change during the runtime, then it is known as dynamic array.

To change size of the array, at run time we use "Resize function"

Jagged Array:-

An array which contains one or more array's within it is known as jagged array. A jagged array is also known as "Array of Array's"



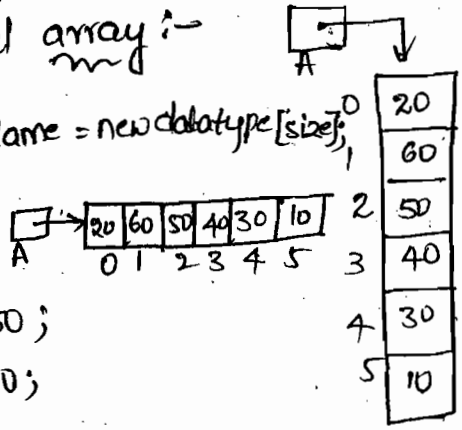
Working with single dimensional array:-

Method 1: Syntax: DataType[ ] ArrayName = new datatype[size];

Ex:- int [ ] A = new int[6];

A[0] = 20; A[1] = 60; A[2] = 50;

A[3] = 40; A[4] = 30; A[5] = 10;



Method 2 : Syntax:- Datatype [ ] ArrayName = new Datatype[size] { initilizing values };

Ex:- int[ ] A = new int[6] {20,60,50,40,30,10}; → count of elements in initialization should be equal to size of the array.

int[ ] A = new int[6] {20,60,50 }; → Raises compilation error

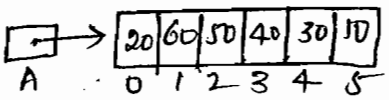
int[ ] A = new int[6] {20,60,50,0,0,0};

Method 3 : Syntax:- Datatype[ ] ArrayName = new datatype[ ] { intilizing elements };

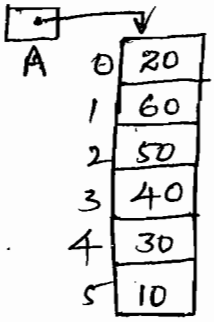
Ex:- int[ ] A = new int[ ] {20,60,50,40,30,10};

Method 4 :- int [ ] A ; null A

A = new int[6] {20,60,50,40,30,10};



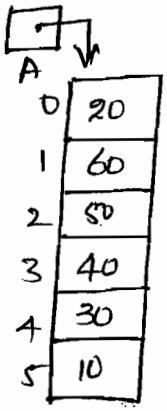
Example to create single dimensional array and print elements on the Screen to User:-




```
namespale CABasics
{ class Example83
  {
    static void Main()
    { int[ ] A = new int[6]
      A[0]=20; A[1]=60; A[2]=50; A[3]=40; A[4]=30; A[5]=10;
      Console.writeline ("Elements of Array are:-");
      for(int i=0; i<=5 or i<6; i++)
       { console.write(A[i] +"  ");
```

```
        Console.Read();
        }
    }
}
```

12) WAP to find sum of the elements of the array



code:-
```
namespace CABasics
{
    class Example84
    {
        Static void Main()
        {
            int[ ] A = new int[6] {20, 60, 50, 40, 30, 10}
            ~~A[0]=20; A[1]=60; A[2]=50; A[3]=40;~~
            int sum=0;
            for(int i=0; i<6; i++)
            {
                Sum=sum+A[i]; (or) sum+=A[i];
            }
            Console.WriteLine("Sum of the elements of array ~~are~~ is:-"+
                                                                sum);
            Console.Read();
        }
    }
}
```

13) WAP to create an array by accepting the elements from the User and print on the Screen to User.

code:-
```
namespace CABasics
{
    class Example85
    {
        Static void Main()
        {
            int[ ] A;
```

```
Console.write("Enter the values of array are:-");
for(i=0; i<6; i++)
  {
    console.writeLine(A[i] + "  ");
  }
  Console.Read();
 }
}
}
```

***

Working with foreach loop:-

Syntax:- foreach(Datatype variable in Array/List/collection Name)

```
{
  statements;
  :
}
```

1. In foreach loop we didn't mention any condition, foreach loop will count number of elements in array, and it will repeat the loop many times.
2. In foreach loop datatype of the loop variable always should be same as the datatype of the array/list/collection.
3. At each iteration of foreach loop, each location value of the array will be copied into loop variable.

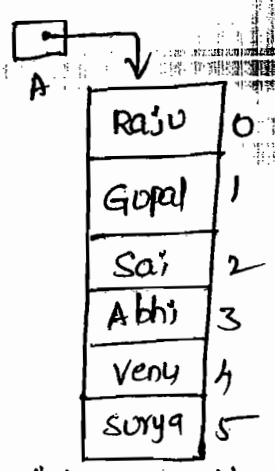
Limitations of foreach Loop:-

1. foreach loop can't be used to refer the elements in reverse order.
2. Foreach loop can't be used to refer only few elements of the array.
3. foreach loop can't be used to initialize the elements or to accept the data from the user.

Example with foreach Loop:-

```
namespace CABasics
{ Class Example86
  {
   Static void Main()
    {int[ ] A = new int[6] {20, 60,50, 40,30,10};
     Console.WriteLine("Elements of array are:-");
     foreach (int i in A)
      {
       Consol e.write( i +"  ");
      }
     console.Read();
    }
  }
}
```

Example to work with string array:-



code:-
```
namespace CABasics
{
  class Example87
  {
    Static void Main()
    {
      String[] A = new String{"Raju", "Gopal", "Sai", "Abhi", "venu",
                              "Surya"};
      Console.WriteLine("Elements of array are:-")
      for(string i in A)
      {
        Console.Write(i + " ");
      }
      Console.Read();
    }
  }
}
```

3. Write a program to find least value within the array
4. Write " " " " highest value within the array.
5. WAP to find average of all values within the array

Solution

Working with Array objects:-

In csharp, when we create any array, array name will become object for array class internally

Array class is avialable in system namespace. This class contains some properties and methods, which are helpful to work with arrays.

Methods with array class object:-

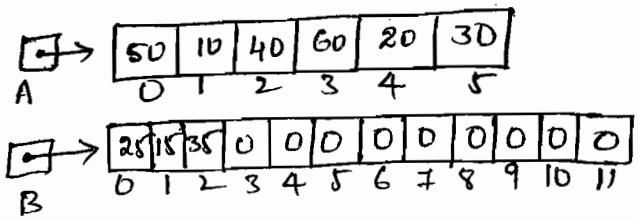
1). CopyTo(Destination Array, int Index)

(2). Max()

(3). Min()

(4). Sum()

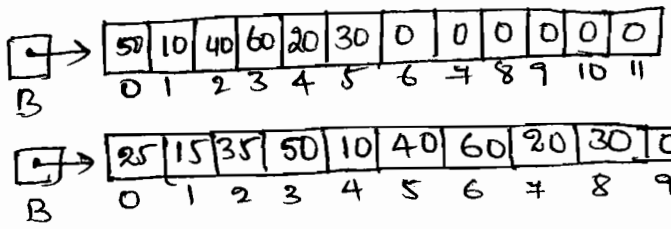
(5). Average()



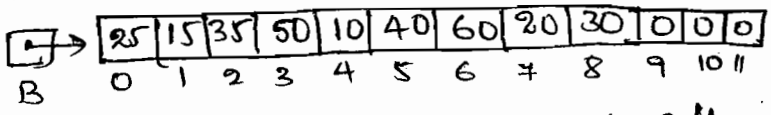
(1) CopyTo with two arguments destination Array, int index:-

This method is used to copy elements of one array into another array

Ex:- A. CopyTo (B, 0) →



A. copy TO (B,3) →



A. copyTo (B,8) → Rises runtime error [Because size n't Sufficient]

(2) Max():- This method will return highest value within the array

Ex:- A.Max ( ) → 60

(3) min():- This method will return

Ex:- A.min( ) → 10

(4) Sum() :- This method will return sum of the element within the array

Ex:- A.Sum ( ) → 210

(4) sum() :- This method will return average of elements within the array

Ex:- A. Average () → 35

properties with Array Object:-

(1) Length

(2) Rank

(1) Length:-

This property will return size of the array

(2) Rank:-

This property will return number of dimensions present in the array.

A.Length → 6	A.Rank → 1
B.Length → 12	B.Rank → 1

Example with Array object methods and properties:-

code:-
```
namespace CABasics
{
    class Example88
    {
        static void Main()
        {
            int [ ] A = new int[6] { 50, 10, 40, 60, 20, 30 };
            int [ ] B = new int[12] { 25, 15, 35, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
            Console.WriteLine("\nElements of Array A are:-");
            foreach (int i in A)
                Console.Write(i + " ");
            Console.WriteLine("\nElements of Array B are:-");
            foreach (int i in B)
                Console.Write(i + " ");
```

```
A.CopyTo(B,3);
C.WL("Elements of array B after copying:-");
foreach(int i in B)
    Console.write(i + " ");
C.WL("Least value in Array A is:-" +A.Min());
C.WL("Highest value in Array A is:-" +A.Max());
C.WL("sum of the Array A Elements is :-" +A.sum());
C.WL("Average of the Array A Elements is:-" +A.Average());
C.WL("Size of Array A is:-" +A.Length);
C.WL("size of Array B is:- " +B.Length);
C.WL("Rank of Array A is :- " +A.Rank);
C.WL("Rank of Array B is :- " +B.Rank);
Console.Read();
    }
  }
}
```

Working with Array Class:-

Array class is available in system namespace. which contains various methods, which are helpful to work with arrays.

Methods with array class:-

i) copy(SourceArray, DestinationArray, int Length)

or

copy(SourceArray, int SourceIndex, DestinationArray, int Destination Index, int Length)

(2). Reverse(ArrayName)

or

Reverse(ArrayName, int Index, int Length)

(3). sort (ArrayName)

or

sort (ArrayName, int Index, int Length)

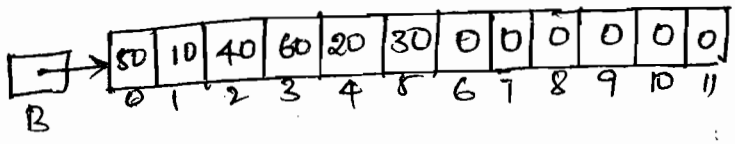
(4). Resize(ref ArrayName, int NewSize)

Note:- If you want to arrange the elements in ascending order, we use method sort. There is no method to arrange the element in descending order. If you want to arrange the elements in descending order, first we sort and reverse all the elements in array.

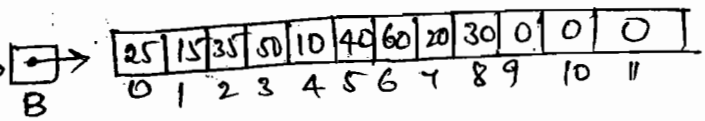
(1). copy method:-

This method is used to copy elements of one array into another array.

Ex:- Array. Copy (A,B,6) →



Array. Copy (A, 0, B, 3, 6) →



Array. Copy (A, 0, B, 8, 6) → Raises runtime Error [Because size n't Sufficient]

(2) Reverse Method:-

This method is used to arrange the elements of array in reverse order.

Ex:- Array.Reverse(A) → A → | 30 | 20 | 60 | 40 | 10 | 50 |
0 1 2 3 4 5

(or)

Array.Reverse(A, 2, 4) → A → | 50 | 10 | 30 | 20 | 60 | 40 |
0 1 2 3 4 5

13). Sort Method:-

This method is used to sort the elements of array. By default elements are arranged in ascending order.

There is no method to arrange the elements in descending order. To arrange the elements in descending order first sort the array and then reverse the array.

Ex:- Array.Sort(A) → A → | 10 | 20 | 30 | 40 | 50 | 60 |
0 1 2 3 4 5

(or)

Array.Sort(A, 2, 4) → A → | 50 | 10 | 20 | 30 | 40 | 60 |
0 1 2 3 4 5

14). Resize Method:-

This method is used to change the size of the array at runtime.

A → | 50 | 10 | 40 | 60 | 20 | 30 |
0 1 2 3 4 5

Ex:- Array.Resize(ref A, 10) → A → | 50 | 10 | 40 | 60 | 20 | 30 | 0 | 0 | 0 | 0 |
0 1 2 3 4 5 6 7 8 9

Array.Resize(ref A, 4) → A → | 50 | 10 | 40 | 60 |
0 1 2 3

Example to methods with Array Class:-

Code:-

```
namespace CABasics
{
class Example89
{
static void Main()
{
int[] A = new int[6]{50,10,40,60,20,30};
int[] B = new int[12]{25,15,35,0,0,0,0,0,0,0,0,0}
C.WL("\nElements of Array A are:-")
foreach (int i in A)
    C.Write(i+" ");
C.WL("\nElements of Array B are:-")
foreach (int i in B)
    C.Write(i+" ");
Array.Copy(A,0,B,3,6);
C.WL("\n Elements of Array B after copying:-");
foreach (int i in B)
    C.Write(i+" ");
Array.Reverse(A);
C.WL("Elements of array A after reversing:-");
foreach (int i in A)
    C.Write(i+" ");
Array.Sort(A)
C.WL("\n Elements of array A after sorting :-");
```

```
foreach(int i in A)
        Console.Write(i +" ");
        Console.Read();
    }
  }
}
```

Example With Resize Method:-

```
namespace CABasics
{
 Class Example 90
  {
   Static void Main()
    {
     int[] A = new int[6] { 50, 10, 40, 60, 20, 30};
     C.WL("Enter Elements of array A are:-");
     foreach(int i in A)
      Console.Write(i +" ");
      Console.Write("\n Enter New Size for the Array:-");
      int S = Convert.ToInt32(Console.ReadLine());
      Array.Resize(ref A, S);
      C.WL("Elements of array A after resizing :-");
      foreach(int i in A)
        Console.Write(i +" ");
        Console.Read();
    }
  }
}
```

→ Exercise:- From above example we use resize function, except filled values in the array are 4 0's. For that places we accept the elements and print the Array on the Screen to user.

Note:- [,] → 1,D Array
[,] → 2D Array
[, ,] → 3D Array
[, , ,] → 4D "
⋮
[, , ... n] → (nth) D Array

} Microsoft Notation.

Working with 2-Dimensional Array:-

A

	0	1	2	3
0	50	60	40	90
1	30	80	65	10
2	55	20	35	70

(1) Syntax:-

DataType[,] ArrayName = new datatype [RowSize, colSize];

Ex:- int [,] A = new int [3,4]

A[0,0] = 50; A[0,1] = 60; A[0,2] = 40; A[0,3] = 90;
A[1,0] = 30; A[1,1] = 80; A[1,2] = 65; A[1,3] = 10;
A[2,0] = 55; A[2,1] = 20; A[2,2] = 35; A[2,3] = 70;

(2) Syntax:-

DataType [,] ArrayName = new datatype [RowSize, colSize]
{{ Row1 Elements--}, {Row2 Elements---}, --- };

Ex:- int[,] A = new int[3,4] { {50, 60, 40, 90}, {30, 80, 65, 10} {55, 20, 35, 70}};

Note:- Count of inner initializers should be equal to Row size and count of elements in each initializer should be equal to column size.

Methods:- Syntax:- DataType[,] ArrayName = new datatype[,] {{Row1 Elements---}, {Row2 Elements---}---};

Ex:- int[,] A = new int[,] {{50, 60, 40, 90}, {30, 80, 65, 10}, {55, 20, 35, 70}};

Example to create a Two dimensional array and print element on the Screen to User:-

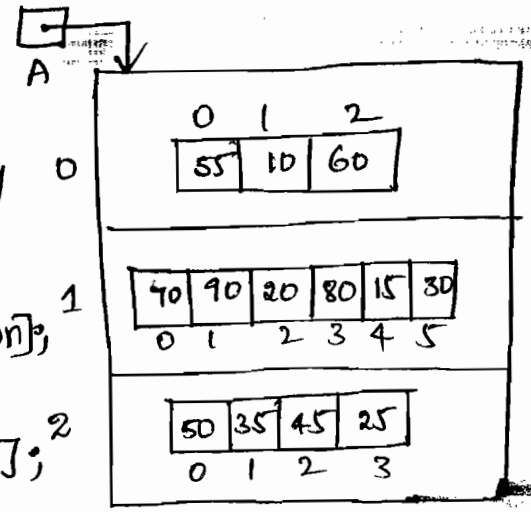
```
code:- namespace CABasics
        {
          class Example91
          {
            static void Main()
            {
              int[,] A = new int[3,4];
              A[0,0]=50; A[0,1]=60; A[0,2]=40; A[0,3]=90;
              A[1,0]= 30; A[1,1]=80; A[1,2]=65; A[1,3]=10;
              A[2,0]=55; A[2,1]=20; A[2,2]=35; A[2,3]=70
              C.WL("Elements of 2D Array are:-");
              for(int R=0; R<=3; R++)
                {
                  for(int C=0; C<=4; C++)
                  {
                    C.Write(A[R,C], +" ");
                  }
```

```
        Console.WriteLine();
      }
      Console.Read();
    }
  }
}
```

Example to print elements of 2D Array using for each loop :-

```
namespace CABasics
{
  class Example92
  {
    static void Main()
    {
      int c=0;
      int [,] A = new int[3,4]
      A[0,0]= 50; A[0,1] = 60; A[0,2] = 40; A[0,3] = 90;
      A[1,0] = 30; A[1,1] = 80; A[1,2] = 65; A[1,3] = 10;
      A[2,0] = 55; A[2,1] = 20; A[2,2] = 35; A[2,3] = 70;
      Console.WriteLine("Elements of 2D array are:-");
      foreach(int i in A)
      {
        Console.Write(i + " ");
        c++;
        if (c==4)
        {
          Console.WriteLine();
          c=0;
        }
      }
      C.RL();
    }
```

Working With Jagged Array's :-



Syntax:-

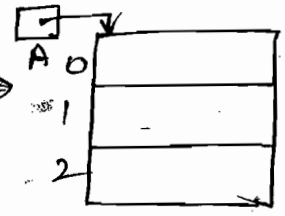
Datatype [Main Array Notation] [Inner Array Notation] ArrayName = new datatype [MainArray ~~Notation~~ size] [Inner Array Notation];

Ex:- int [ ] [ ] A = new int [3] [ ];

A[0] = new int[3] { 55, 10, 60};

A[1] = new int[6] { 70, 90, 20, 80, 15, 30};

A[2] = new int [2] { 50, 35, 45, 25};



Example to create a jagged array and print the elements on the Screen to user:-

```
namespace CABasics
{ class Example 93
  {
    static void Main()
    { int [ ] [ ] A = new int [3] [ ];
      A[0] = new int [3] { 55, 10, 60};
      A[1] = new int [6] { 70, 90, 20, 80, 15, 30};
      A[2] = new int [4] { 50, 35, 45, 25};
      C.WL("Elements Of Jagged Array are:-");
      for (int R=0; R<A.length; R++)
      {
        for (int C=0; C<A[R].length; C++)
        {
          C.Write(A[R][C] + " ");
        }
        C.WL();
```

R=0 → A[0] → 55 10 60

R=1 → A[1] → 70 90 20 80 15 30

R=2 → A[2] → 50 35 45 25

A[R][C] like

⇓

A[0][0] → 55

A[1][3] → 80

```
        C.Read();
    }
  }
}
```

Use of jagged arrays:

→ Jagged array's are used in gaming programming

Ex:- all card games

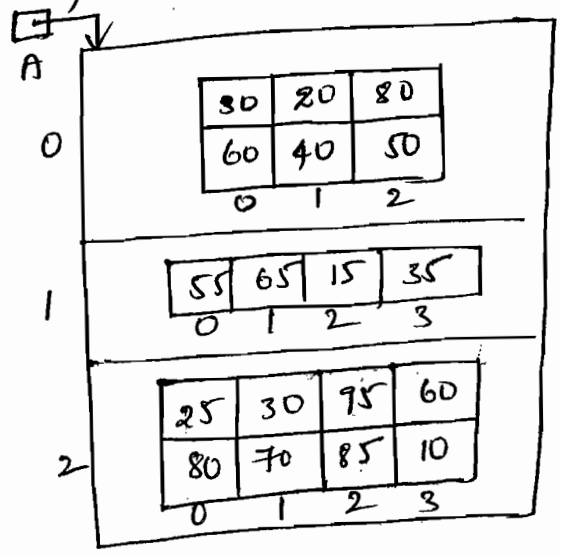
In jagged arrays where row ~~size are~~ fixed and column size are n't fixed.

Jagged Array by using for each loop:-

```
namespace CABasics
{
  class Example95
  {
    Static Void Main()
    {
      int[ ] A = new int[3][ ];
      A[0]=new int[3]{ 50,10,60};
      A[1] = new int[6]{70,90,20, 80,15,30};
      A[2]=new int[4] { 50,35, 45,25};
      foreach (int[ ] i in A)
        {
          foreach (int j in i)
            {
              Console.write(j + " ");
            }
          Console.writeLine( );
        }
      Console.Read( );
    }
  }
}
```

03/11/2014

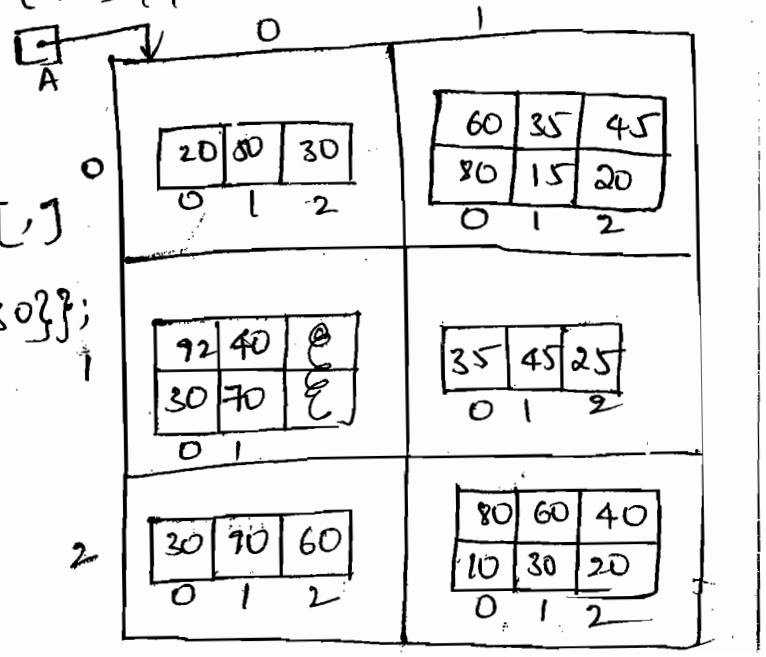
3. To create Array for the below, But don't print elements.



code:- namespace CABasics

```
{
  class Example96
  {
    static void Main()
    {
      int[ ][,] A = new int[3][,];
      A[0] = new int[2,3]{{30,20,80};{60,40,50}}
      A[1] = new int[1,4]{{55,65,15,35}}
      A[2] = new int[2,4]{{25,30,95,60};{80,70,85,10}};
```

4. To create array for this




```
int[,][,] A = new int[3,2][,]
A[0,0] = new int[1,3]{{20,50,80}};
```

OBJECT ORIENTED PROGRAMMING [OOP]

Run. Notes

4 important pillar's or OOP:-

① Abstraction:-

Hiding the implementation but providing the service or result.

There are 2 types of abstraction

(i) functionality abstraction

(ii) Data abstraction

② Encapsulation:-

Binding member variables ~~are~~ of a class hidden along with ~~encapsulation~~ member functions.

③ polymorphism:-

showing different forms

poly means → many

morphism means → many forms.

polymorphism 2 types

(i) static polymorphism / compiletime polymorphism / early Binding

↳ achieved by two things (i) operation overloading

(ii) function overloading

(ii) Dynamic polymorphism / runtime polymorphism / Late Binding

↳ achieved by function overriding

④ Inheritance:-

↳ main purpose of inheritance is to provide reusability and additional enhancement

04/11/2014 Before object orient programming there was procedure oriented programming. In procedure oriented prog. there was a greater drawback available when a problem is solved. once the problem is enhanced or redefined, the existing solution cannot be reused.

→ To overcome this drawback object ~~oriented~~ programming is introduced

→ The backbone of OOP is "class and object."

class:-

A class is collection of things that possesses common features or similarities.

Ex:- Birds is a class which contain common features like every Bird has two legs, two wings, every bird has some colour and some weight. Every bird can fly, etc...

Object:-

An object is used to represent the class. without object It is impossible to represent the class.

An object is known as an instance of the class.

Ex:- parrot is an object to the birds class.

A class is also can be ~~class~~ said as blue print about what we want to prepare

If you want to represent a car in object oriented programming

We write like

```
class car
{ Engine, Body, wheels, Doors, Gears, Fuel, Break, etc...
  Void Manufacture()
  {
   !
  }
  Void Fit Electricity()
  {
   !
  }
  Void paint()
  {
   !
  }
}
```

Here Engine, body, wheels, doors, etc... are the member variables of the class and manufacture, fit electricity and paint are member functions of the class.

Member functions will always provide the functionality that can be provided within the class.

Features of object oriented programing:-

OOP will support following features which are called pillar's of OOP. They are

(1) Abstraction (2), Encapsulation (3), polymorphism

(4) Inheritance

(1). Abstraction :-

Abstraction is the process of hiding the implementation but providing the service or result.

There are 2 types of abstraction available.

(i). Data abstraction

(ii). functional abstraction

(2) Encapsulation:-

Encapsulation is the process of binding member variables of a class along with member functions.

(3). polymorphism:-

The word polymorphism has been derived from greek, where poly means many, morph means behaviours or forms.

Same function or operator will show different result or will behave in different ways, when number of arguments or ~~are~~ datatypes of arguments are changed

Types of polymorphism:-

There are 2 types of polymorphism,

(1) Static polymorphism

(2). Dynamic polymorphism.

Types of polymorphism

- Static / compile time polymorphism
  (or)
  Early Binding
  ↳ Achieved by using
    (1). operator overloading
    (2) Function overloading

- Dynamic / Runtime polymorphism
  (or)
  Late Binding
  ↳ Achieved by using
    (1) Function overriding

(4) Inheritance:-

Inheritance is the process of creating a new class from already avialable class. Already avialable class is known as parent class or base class. Newly created class is known as derived class or child class.

Within the inheritance process always derived class will get all the features of parent class or base class.

The main purpose of inheritance is reusability and to providing enhancements whenever required.

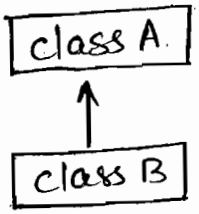
Types of Inheritance:-

There are following types of inheritance supported in object oriented programming.

(a) Single inheritance
(b). Multilevel inheritance
(c). Multiple inheritance
(d). Hybrid inheritance
(e). Hierarchial inheritance

(a). Single inheritance:-

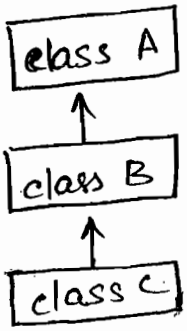
creating a single new class from a single base class is known as single inheritance.



Here classA is known as parent class or base class and Class B is known as derived class or child class.

(b) Multilevel inheritance:-

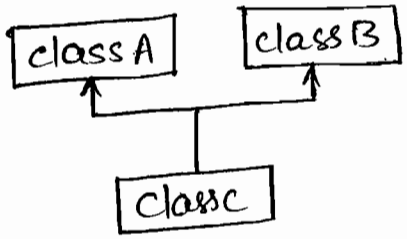
creating a new class from already derived class is known as multilevel inheritance.



Here class A is known as parent class for class B, class B is derived ~~from~~ for class A. At the same time class B is base class for class C, and class C is derived class ~~from~~ for class B. class B is also can be called as intermediate base class.

(c) Multiple inheritance:-

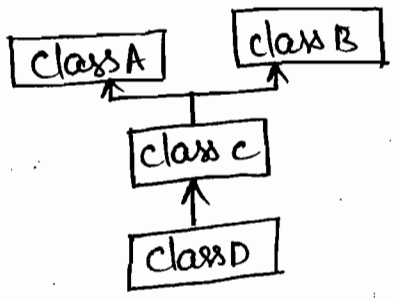
Creating a new class from two or more classes at base level, is known as multiple inheritance



Here class A and class B are parent classes for class C and class C is derived class for both class A and class B.

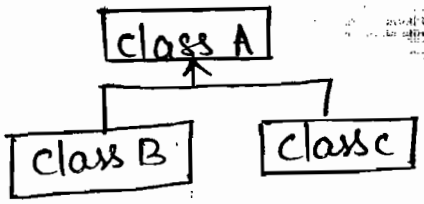
(d). Hybrid inheritance:-

This is the combination of both multiple and multi-level inheritances.



(e) Hierarchial inheritance:-

Creating two or more classes from a single base class is known as hierarchical inheritance.



Syntax to create a class in C# :-

```
AccessModifier class ClassName
{
    code
    :
}
```

Access-Modifier can be anyone from

(a) private
(b). protected
(c). internal.
(d). protected internal
(e) public.

* Default AccessModifier of the class is "internal"

In c# a class contain 10 members.

(a) Datafields
(b). Functions
(c). Constructors
(d). destructors
(e). properties
(f). Indexes
(g). Events
(h). Constants
(i). operator
(j). Nested type

* Default accessmodifier of any member of the class is "private"

1a) Datafields:-

Datafields in the class are responsible to store the data related to the class within the class.

Except datafields no other member of class is responsible to store the data.

To create a datafield within the class we use following syntax

Syntax:- AccessModifier Datatype datafieldName.

Ex:-
```
public int EmpID;
protected String Ename;
internal int age;
private double sal;
```

1b) Functions:-

Method:-

Method is a reusable piece of code which is used to perform required task

A Method can be two parts

1a) procedure

1b) Functions.

(a) procedure:-

procedure is a method which will "~~never~~ return any value to the calling function"

As procedure doesn't return any value of class and procedure doesn't have any return type

Ex:- Add (int a, int b)

```
{
  :
}
```

To call the procedure we write a code like Add (10,20);

(b) Functions:-

A function is a method, which will "return always a single value" to the calling function.

As the function return single value to the calling place the function mandatory should have return type.

Ex:- int add (int a, int b)

```
{
  :
  return a+b;
}
```

To call these function we write a code like, int c = add(10,20)

Here function is returning the value, so we mentioning value as "int".

If we want to create function without returning any value to the calling function use the return type "void"

Ex:- void add(int a, int b)

{

}

To call these we write the code like add(10,20).

* c# doesn't support procedure and just support function only. so for any method in c# "mentioning return type is mandatory".

When we write the main method or main function then we mention the return type as "void". Because we are not returning any value to main and c# doesn't support procedure. Function mandatory should have return type.

→ In c# main method have 2 return types.

(1). void → not return any value

(2). int → to return successful or unsuccessful Expression.

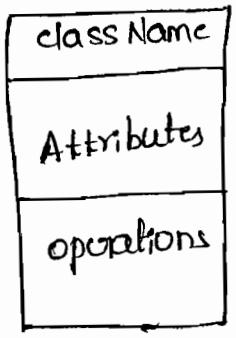
→ Some people also called a function written inside the class is "return".

06/11/12 Diagramatical Representation of a class:-

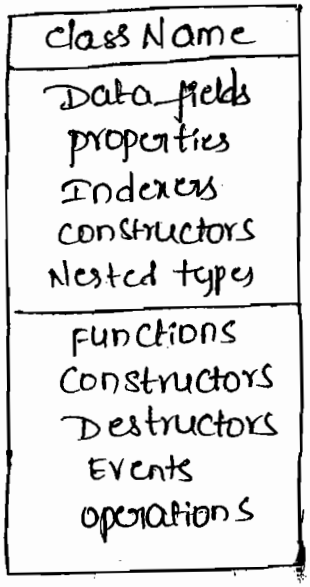
A class is represented by the rectangle, which is divided into 3 parts.

First part always will contain class name, second part contains attributes and 3 part contains operations.

In UML



In C#



UML → Unified Modeling language

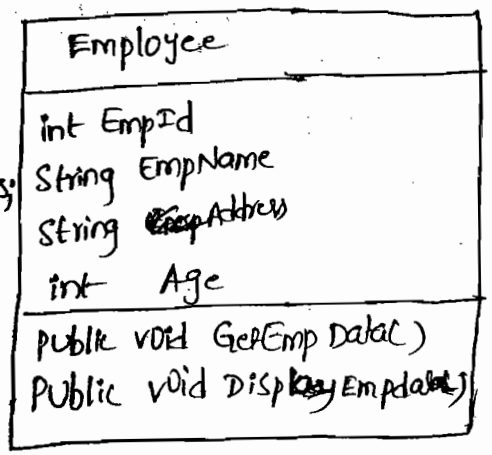
* Java doesn't Support Indexers.

Syntax to create object of the class:-

```
className objectName = new className([args values])
                           ↓
                       Constructor Name
```

Example to create and consume a class:-

```
namespace cAclassExamples
{ class Employee
  {int EmpId, Age; String EName, Address;
   public void GetEmpData( )
   { c.W(" Enter Employee Details:-");
```



```
EmpId = Convert.ToInt32 (C.RL());
EName = C.RL();
Address = C.RL();
Age = convert.ToInt32 (C.RL());
}
public void DispEmpData()
{
  C.WL("Employee Id is:-" + EmpId);
  C.WL("Employee Name is:-" + EName);
  C.WL("Employee Address is :-" + Address);
  C.WL("Employee age is :-" + Age);
  }
}
class Example1
{
  Static void Main()
  {
    Employee Emp1 = new Employee(); → According to syntax to create object of a class
    Employee Emp2 = new Employee();
    Emp1.GetEmpData();
    Emp2.GetEmpData();   } → syntax:- ObjectName.MemberName (or) ClassName.MemberName
    Emp1.DispEmpData();
    Emp2.DispEmpData();
    C.Read();
  }
}
}
```

1. What happens when an object is created to class? Why use of "New" keyword?

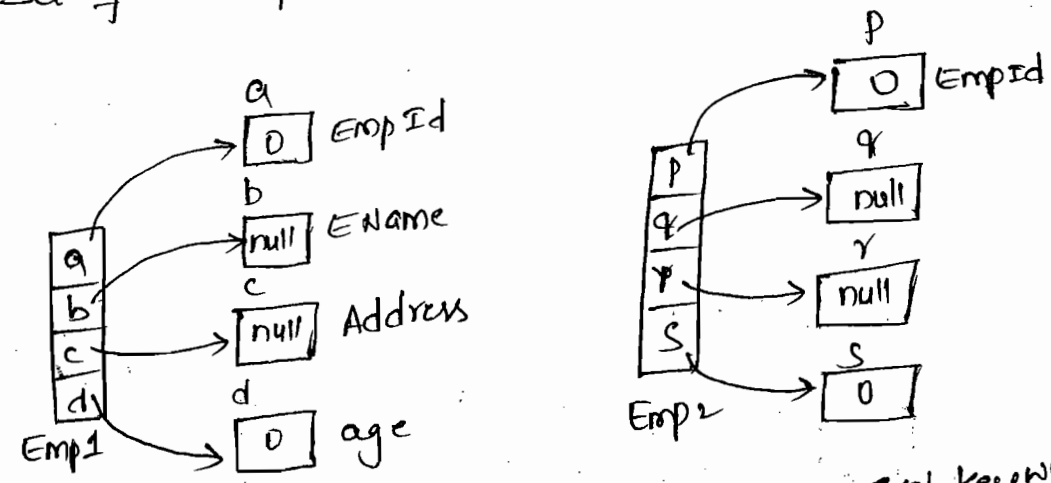
Ans:- New keyword is responsible to invoke the constructor of the class and when object is created to class This constructor will perform 3 tasks simply,

(1) memory is alloted to the data fields.

(2) Default values are stored into the data fields.

(3). References of these data fields are maintained or copied from the object.

so Here we created 2 objects. For 2 objects separate set of data fields will be there and we find it like



2. What happens if you don't use new keyword when creating object of the class.

Ans:- If you don't use new keyword when creating an object of the class, memory will n't alloted to data fields and object will have null reference like

Employee Emp3;

null

Emp3.

07/11/2014

3. Explain about how functional abstraction is implemented in the above program.

Ans:- In the above program when we are calling Emp1. GETEmp Data Emp1 reference datafields will bind to get ~~emp1x~~ function data.

Similarly when we are calling emp2. GET emp data, Emp2 ~~dat~~.

referenced datafield will bind to GetEmp2 data function

originally this binding will takeplace using "this object" or "this keyword"

4. What is "this"?

Ans:- This is an object to the current class within the same class.

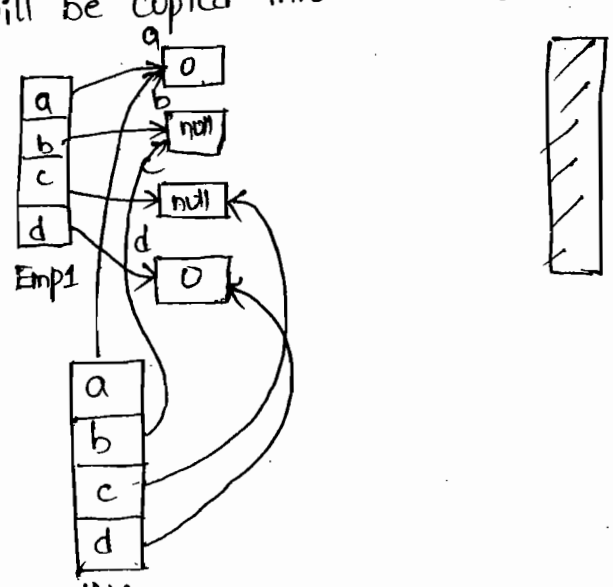
5. How this object will work?

In the above program we can write the class code like

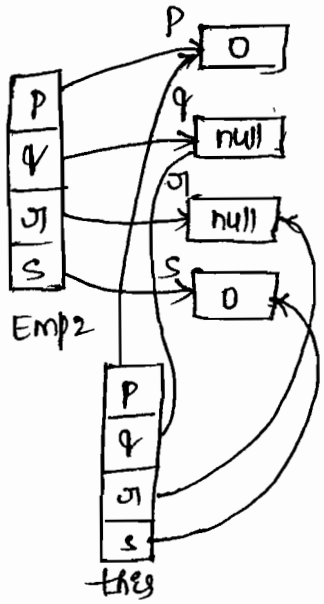
```
public V. GetEmpData ( ) { C.W ("Enter Employee Details:-");
    this.Empid = Convert.ToInt32(Console.ReadLine());
    this.EName = Console.ReadLine();
    this.Address = Console.ReadLine();
    this.Age = Convert.ToInt32(Console.ReadLine());
}
```

```
public void dispEmpData()
{ C.WL("Employee id is:-"+this.Empid);
  C.WL("Employee Name is:-"+this.EName);
  C.WL("Employee Address is:-"+this.Address);
  C.WL("Employee Age is:-"+this.Age);
}
}
class Example1
{ static void Main(Strg[] Args)
  { Employee Emp1 = new Employee();
    Employee Emp2 = new Employee();
    Emp1.GetEmpData();
    Emp2.GetEmpData();
    Emp1.DispEmpData();
    Emp2.DispEmpData();
    Console.Read();
  }
}
```

when we are calling Emp1.GetEmpData, Emp1 referenced data fields will be copied into this object. like.



When we are calling Emp2.GetEmp2Data, Emp2 references will be copied into this object like



Here with which object, we call any functions that object functions are copied into "this" object, so that object referenced data fields are binding to that function. This is how Encapsulation is implemented

6. Explain about how functional abstraction is implemented.

Ans:- In the above Example we have taken 2 functions

i.e., (1) GetEmpData
(2) DisplayEmpData In Employee class.

Let us assume that if there are no functions, then above Example code will be like,

```
{ namespace cAclassExamples
  { class Example
    { public int EmpId, Age;
      public string EName, Address;
```

```
class Example1
{
  s v Main()
  {
    Employee Emp1 = new Employee();
    ~~Employee Emp2 = new Employee();~~
    C.write("Enter Employee details:-");
    Emp1.EmpId = C.ToInt32(C.RL());
    Emp1.EName = C.RL();
    Emp1.Address = C.RL();
    Emp1.Age = C.~~Read~~ ToInt32(C.RL());
    C.WL("Employee Id is :-" + Emp1.EmpId);
    C.WL("Employee Name is :-" + Emp1.EName);
    C.WL("Employee Address is :-" + Emp1.Address);
    C.WL("Employee Age is :-" + Emp1.Age);
    C.Read();
  }
}
```

Here if there are no functions, in Employee class then Example1 class is getting to know about the implementation of Employee class.

If there are functions avialable, then Example1 class will n't know the implementation or functionality of Employee class. Here we are hiding the functionality or implementation of employee class from other classes by writing functions. so this is called functional abstraction.

Ex:- create a class and writing the code for calculating Employee salary.

EmpSalary
int EmpId String EName double Basic double DA → 40% of Basic double Hra → 30% of Basic double Gross
public void GetData() public void calculate() Public void DispData()

code:-

```
namespace CAClassExamples
{ class Example
{ int EmpId;
  String EName;
  double Basic, DA, Hra, Gross;
  public void GetData()
  {
    C.WL("Enter Employee details:-");
    this.EmpId = Convert.ToInt32 (C.RL());
    this.EName = Convert.ToString (C.RL());
    this.Basic = Convert.ToDouble (C.RL());
    this.DA = Convert.Double (C.RL());
    this.Hra = Convert.Double (C.RL());
  }
  public void calculate()
  {
    this.DA = this.Basic * 0.4
    this.Hra = this.Basic * 0.3
    this.Gross = this.Basic + this.DA + this.Hra;
  }
  public void DispData()
  {
    C.WL("Employee Id is :-" + this.EmpId);
    C.WL("Employee Name is :-" + this.EName);
    C.WL("Employee Basic is :-" + this.Basic);
    C.WL("Employee DA is :- " + this.DA);
```

```
C.WL("Employee Hra is :- " + this.Hra);
C.WL(" Employee Gross is :- " + this.Gross);
}
}
class Example2
{
static void Main()
{
Empsalary obj1 = new Empsalary();
obj1.GetData();
obj1.calculateData();
obj1.DispData();
Console.ReadLine();
}
}
}
```

Ex3:- create a class and write the code to accept details and print on the Screen to user.

code:-

```
namespace CAclassExamples { class Book
{
int BId; String BName, AName, PName;
Double MRP;
public void GetBData()
{
this
console.WriteLine("Enter Book Details:-");
this.BId = Convert.ToInt32(console.ReadLine());
this.BName = Console.ReadLine();
this.AName = Console.ReadLine();
```

BOOK
int BId String BName String AName String PName double MRP
public void GetBdata() Public void DispBdata()

```
this. PName = Console. ReadLine();
this. MRP = Convert. ToDouble( Console. ReadLine());

public void DispBData()
{
console. WriteLine(" BId is:-" + this. BId);
console. WriteLine("BName is:-" + this. BName);
Console. WriteLine("AName is:-" + this. AName);
Console. WriteLine("PName is:- " + this. PName);
Console. WriteLine(" MRP is:-" + this. MRP);
}
class Example3
{
static void Main()
{
Book B1 = new Book();
B1. GetBdata();
B1. DispBdata();
Console. ReadLine();
}
}
}
```

Ex:- Create a class to accept the length and breath of a rectangle to find the Area and perimeter of a rectangle.

Rectangle
int length int Breadth int Area int perimeter
public void Accept() public void calculate() public void print()

Ex:- create a class and write the code to accept Student details calculate total marks, Result print on the Screen to user.

Student
int RNO String SName int M1 int M2 int M3 int M4 int Total String Result
public void Getdata() public void calculate() public void Dispdata()

Ex:- namespace CAClassExamples

```
{
  class Rectangle
  {
    int length, Breadth, Area, perimeter;
    public void Accept()
    {
      console.writeLine(" Enter length and breadth:-");
      this.length = Convert.ToInt32(C.RL());
      this.breadth = Convert.ToInt32(C.RL());
      this.Area = Convert.ToInt32(C.RL());
      this.perimeter = Convert.ToInt32(C.RL());
    }
    public void calculate()
    {
      this.area = this.length * this.breadth;
      this.perimeter = 2 (this.length + this.breadth)
    }
    public void print()
    {
      console.writeline (" length is :-" + this.length);
      conside.writeLine( " breadth is :-" + this.breadth);
      conside.writeline(" Area is :-" + this.Area);
      conside.writeline( " perimeter is :-" + this.perimeter);
    }
```

```
class Example4
{
  Static void Main()
  {
    Rectangle obj1 = new Rectangle();
        obj1.Accept();
        obj1.calculateData();
        obj1.printData();
        Console.Read();
  }
}
}
```

Ex5:-

```
namespace CAclassExample
{ class Student
  {
    int RNO, M1, M2, M3, M4, Total; String SName, Result;
    public void GetData()
    {
      Console.WriteLine("Enter SName, marks and RNO:-");
      this.RNO = Convert.ToInt32 (C.RL());
      this.M1 = Convert.ToInt32 (C.RL());
      this.M2 = Convert.ToInt32 (C.RL());
      this.M3 = Convert.ToInt32 (C.RL());
      this.M4 = Convert.ToInt32 (C.RL());
      this.Total = Convert.ToInt32 (C.RL());
      this.SName = Console.ReadLine();
      this.Result = Console.ReadLine();
    }
```

```
public void calculate()
{
  this.Total = M1+M2+M3+M4;
  {
   if (M1>=35 && M2>=35 && M3>=35 && M4>=35)
  this.Result = pass;
    else
  this.Result = fail;
    }
 }
public void DispData()
 {
  Console.ReadLine("Total is :-" +this.Total);
  Console.ReadLine(" Result is :- " +this.Result);

 }
class Example5
 {
  static void Main()
   {
    Student obj1 = new Student();
            obj1.GetsData();
            obj1.calculateData();
            obj1.DispData();
            Console.ReadLine();
      }
   }
 }
```

08/11/2014

Working with Constructors:-

A constructor is a special member function of a class which is invoked automatically, when an object to the class is created.

In general, a constructor name should be same as class name.

→ A constructor doesn't return any return type even void also.

→ There is no ristriction to write particular code in constructor based on requirement, we can write any type of code.

→ When a constructor is invoked, 3 tasks will be performed implicitly

(i), Alloting Memory to the data fields.

(ii) storing the default values into the data fields

(iii) Maintaining reference of these data fields, from the object of the class.

After performing this tasks any other code that we write in the constructor will be executed.

A constructor is used "to keep something ready for the object," when an object to the class is created.

* Always constructor will same as class name.

Types of constructor's:-

- Types of constructors
  - Instance constructors
    - Default constructors
      - User defined default constructor
      - System defined default constructor
    - parameterized constructors
    - copy "
    - private "
  - Non-Instance constructors → static constructors

User defined default constructor:-

This constructor doesn't have any parameters and is created by the programmer.

In general, any type of code can be written within this constructor,

→ 1. Example with User defined default constructor:-

```
namespace CAconstructors
{ class Example1
  { int EmpId, Age; String EName, Address;
    public Employee1()
    { this. EmpId = 101;
      this. EName = "Raju";
      this. Address = "Hyderabad";
      this. Age = 20;
    }
```

Employee 1
int EmpId String EName String Address int Age
Public Employee1() Public void DispEmpdata()

```
public void DispEmpData()
{
    C.WL("Employee Id is :-" + this.EmpId);
    C.WL("Employee Name is :-" + this.EName);
    C.WL("Employee Address is :-" + this.Address);
    C.WL("Employee Age is :-" + this.Age);
}

}
Class Example1
{ static void Main()
  {
    Employee1 obj1 = new Employee1();
    Obj1.DispEmpData();
    Console.Read();
  }
}
}
```

System Defined Default Constructor:-

When there is no constructor present within the class, run time will create its own constructor and perform 3 Tasks.

This constructor created By runtime is known as system defined default constructor.

→ Example with System Defined Default constructor:-

```
namespace CAconstructors
Class Example2
{ int EmpId, Age; String EName, Address;
```

Employee2
int EmpId String EName String Address int Age
public void DispEmpData()

```
public void DispEmpData()
{
    C.WL("Employee Id is :- " + this.EmpId);
    C.WL("Employee Name is :-" + this.EName);
    C.WL("Employee Address is :-" + this.Address);
    C.WL("Employee Age is :-" + this.Age);
}
class Example2
{
    S.V Main()
    {
        Employee2 obj1 = new Employee2();
        Obj1.DispEmpData();
        C.Read();
    }
}
}
```

Disadvantage using Default constructors when initializing fixed values:-

Any Number of objects may be created to the class all objects will store same data into their data fields. To overcome this drawback we depend on parameterized constructor.

parametized Constructor:-

This constructor is used to store separe set of values, for each object of the class created.

parameterized constructor have one or more parameters.

10/11/2017 Example with parameterized Constructors:-

code:-

```
namespace CAConstructors
{ class Employee3
{int EmpId, String EName, Address;
  Age;
  ~~int Age~~;
  public Employee3(int Id, String S1, String S2,
                   int Age)
{
  this.EmpId = ~~raj~~ Id;
  this.EName = 'Raju'; S1;
  this.Address = 'Hyderabad' S2;
  this.Age = '20' Age;
}
public void DispEmpData( )
{
    C.WL(Employee Id is :-" + this.EmpId);
    C.WL(Employee Name is :-" + this.EName);
    C.WL(Employee Address is :-" + this.Address);
    C.WL(Employee Age is :-" + this.Age);
}
class Example3
{
  Static void Main( )
  {Employee3 obj1 = new Employee3(101, 'Raju', 'Hyd', 26);
   Employee3 obj2 = new Employee3(102, 'Gopal', 'BAN', 28);
      obj1.DispEmpData();
      obj2.DispEmpData();
      Console.Read();
  }
}
```

Employee3
int EmpId String EName String Address int Age
public Employee3 (int Id, String S1, String S2, int Age) public void DispEmpData()

In the above example, If we would like to create object for the class Employee3 like,

```
Employee3 obj3 = new Employee3();
```

It is n't possible. Rather class should contain a constructor rather it raises compilation error like 'CAconstructors.Employee3' does not contain a constructor that takes 0 arguments.

To overcome this; we should create a constructor in the class Employee3, without any arguments like

```
public Employee3()
{
  this.EmpId = 103;
  this.EName = "Sai";
  this.Address = "Chennai"
  this.Age = 25;
}
```

If we would like to create an another object, to the class Employee3 like

```
Employee3 obj4 = new Employee3(104, "Venu");
```

It is n't possible to create object in this way rather it raises a compilation error like 'CAconstructors.Employee3' does not contain a constructor that takes 2 arguments.

To overcome this we should create one more constructor within the class Employee3 like

```
public Employee3 (int Id, string S1)
{
  this.EmpId = Id;
  this.EName = S1;
}
```

*** From the above things, we can understand that,

(i) A class can contain any Number of constructors and

(ii) A constructor can be overloaded

Copy Constructor:-

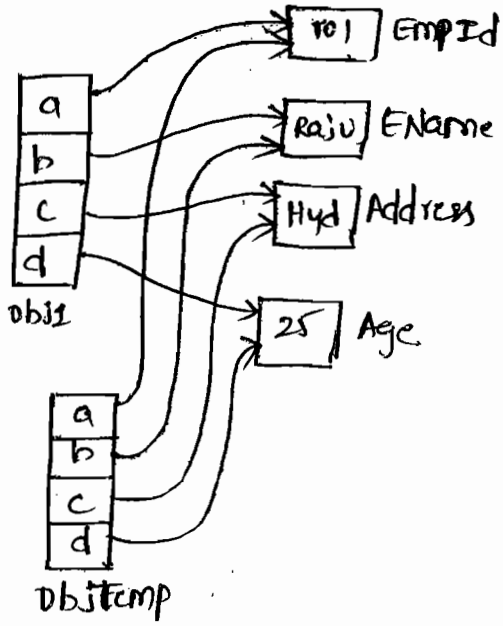
A copy constructor is used, to copy the data of an Existing object into Newly created object

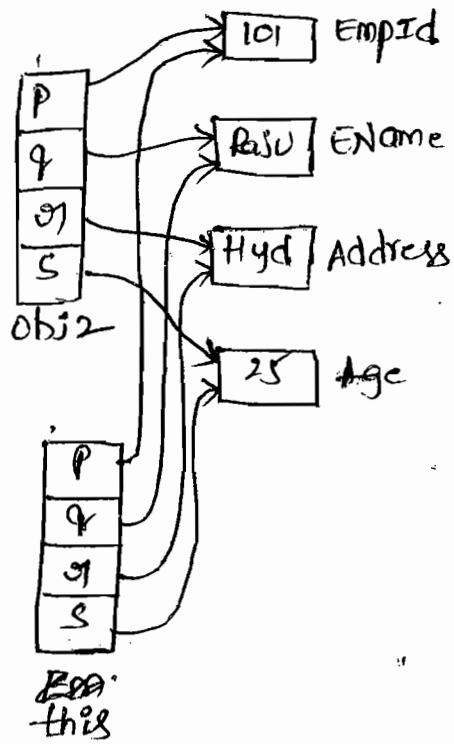
→ Example with copy constructor:-

```
namespace CAConstructors
{
  class Employee4
  {
    int EmpId, Age; string EName, Address;
    public Employee4()
    {
      Console.Write("Enter Employee Details:-");
      this.EmpId = C.ToInt32((C.RL));
      this.EName = C.RL();
      this.Address = C.RL();
      this.Age = C.ToInt32 (C.RL());
```

Employee4
int EmpId String EName String Address int Age
public Employee4() public Employee4(Employee4 obj Temp public void DispEmpData()

```
public Employee4(Employee4 objTemp)
{
  this.EmpId = objTemp.EmpId;
  this.EName = objTemp.EName;
  this.Address = objTemp.Address;
  this.Age = objTemp.Age;
}
public void DispEmpData()
{
  C.WL("Employee Id is:-"+ this.EmpId);
  C.WL("Employee Name is:-"+ this.EName);
  C.WL("Employee Address is:-"+ this.Address);
  C.WL("Employee Age is:-"+this.Age);
}
class Example4
{
  static void Main()
  {
    Employee4 obj1 = new Employee4();
    Employee4 obj2 = new Employee4(obj1);
    obj1.DispEmpData();
    obj2.DispEmpData();
    C.Read();
  }
}
}
```





Employee4 obj3 = Obj2)

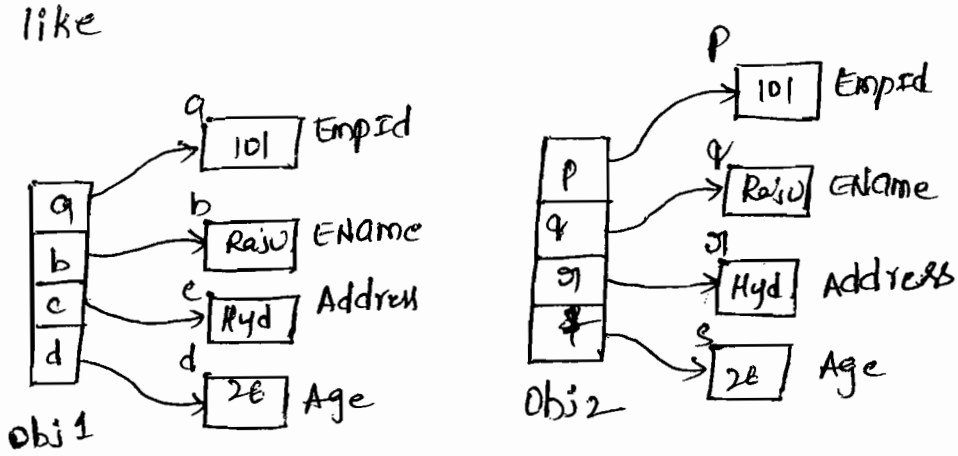
In the above example, we can also write Employee4 obj3 = obj1;

To copy the data of object. But the difference b/w copy constructor and copying the data using assignment operator is Deep copy and shallow copy.

If we copy the data using constructor like

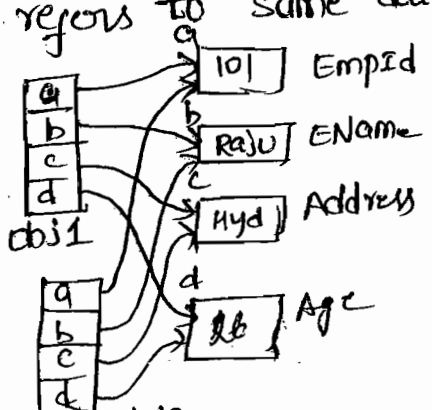
~~Employee4~~ obj2 = new Employee4 (obj1)

In this case, for both objects separate set of memory is alloted to the data fields. For obj1 and obj2 we find memory like



In case if you are copying data using assignment operator like Employee4 obj3 = obj1;

For obj3 separate set of datafield's will n't be alloted rather obj3 refers to same datafields of obj1 like



copying the data one object into other using copy constructor is called "deep copy"

copying the data of one object into other object using assignment operator is known as "shallow copy"

In case of deep copy for every object separate set of datafields are maintained,

In case of shallow copy, for all objects same set of data fields will be maintained.

Note:- At any point of time if you don't use new keyword separate set of datafields will n't be created, For the object. Rather object will be refering to data field of other object, which is called a reference

Working with private Constructors:-

A constructor whose accessability is private is known as private constructor.

If a class contains all private constructors, then it is n't possible to create object for the class from outside the class. But we can create object within the same class and consume.

If a class contains some public constructors and some private constructors, then we can create object to the class from outside the class using public constructors.

where private constructors are used in real time. private constructors in realtime are using in "remoting options."

→ Example with private constructors :-

Employee5
int EmpId String EName String Address int Age
Public Employee5 Public void DispEmpData()

code:-

```
namespace CAConstructors
{
  class Employee5
  {
    int EmpId, Age; String EName, Address;
    private Employee5()
    {
      Console.Write("Enter Employee Details:-");
      this.EmpId = C.ToInt32(C.RL());
      this.EName = C.RL();
      this.Address = C.RL();
      this.Age = C.ToInt32(C.RL());
    }
```

```
public void DispEmpData()
{
    C.WL("Employee Id is:-" + this.EmpId);
    C.WL("Employee Name is:-" + this.EName);
    C.WL("Employee Address is:-" + this.Address);
    C.WL("Employee Age is:-" + this.Age);
}
Static void Main()
{
    Employee5 obj1 = new Employee5();
    obj1.DispEmpData();
    C.R();
}
}
}
```

Static Constructor and Static Data fields:-

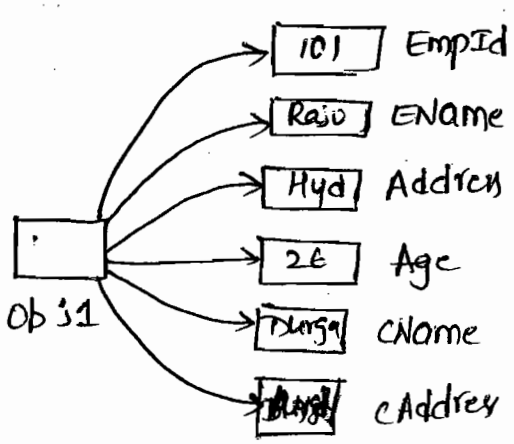
Why Static Data fields:-

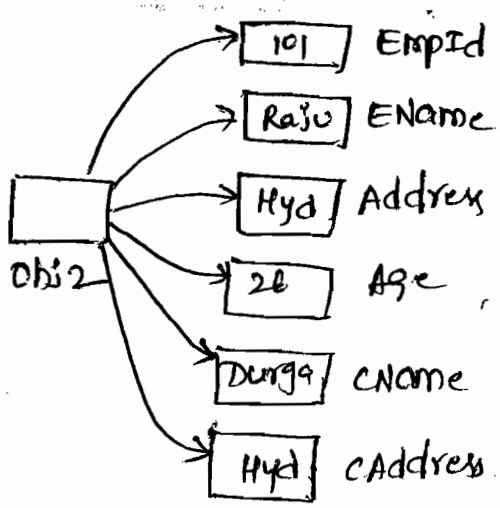
Assume that, we have a class like

Employee5
int EmpId String EName String Address int Age ~~Static~~ String CName String CAddress

→ To consume this class we create an object

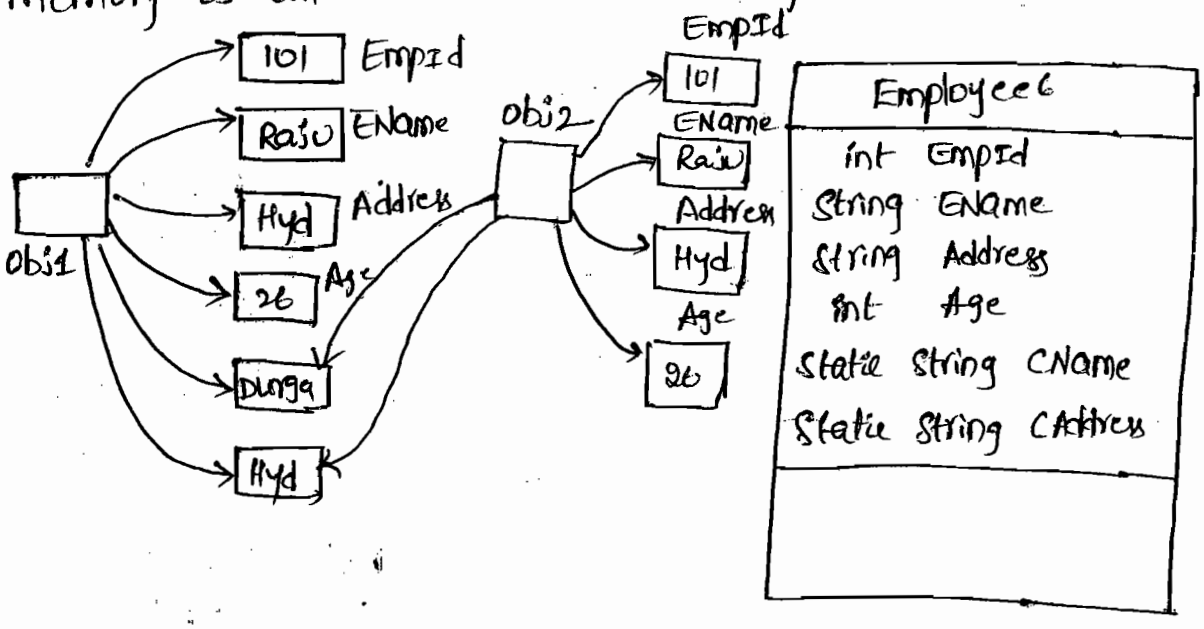
→ If there are we create an object, memory is alloted separately to the ~~to~~ data fields,





But if there are 100 employees, if we create 100 objects, then CName, CAddress i.e., company Name and Company Address. 100 times memory is alloted separately. Originally which is n't Required, and will waste the memory.

To overcome this drawback, we depend on static data fields. If we make CName, CAddress is static the common memory is alloted to these data fields for all objects like



→ A static constructor is used to initialize a static members of the class.

→ A static constructor will be executed only once that to for the first object created to the class. For remaining objects static constructor will n't called for execution.

→ Default Accessability of static constructor is "public"

→ In static and non-static constructors, static constructor will be called first, than non-static constructor will called later.

→ In static constructors it is n't possible to initialize non-static data fields. It raises compilation Error.

→ In non-static constructors, we can initialize static constructor, but they loose their static Nature.

Example with static Constructor

Employee6
int EmpId String EName String Address int Age Static String CName Static String CAddress
public Employee6() Static Employee6() public void DispEmpDetails()

12/11/2017

```
namespace CAConstructors
{
class Employee6
{
int EmpId, Age; String EName, Address;
Static String CName, CAddress;
public Employee6()
{
Console.WriteLine("Enter Employee details:-");
this.EmpId = Convert.ToInt32(C.RLL());
this.Age = Convert.ToInt32(C.RLL());
this.EName = Console.ReadLine();
this.Address = Console.ReadLine();
}
public Static Employee6
{ c.write("Enter Company Name & Address:-");
CName = C.RLL();
CAddress = C.RLL();
}
public void DispEmpData()
{
c.WL("EmpId is:-" +this.EmpId);
c.WL(" Age is :- " +this.Age);
c.WL("EName is:-" +this.EName);
c.WL("Address is:-" + this.Address);
c.WL("Company Name is:-" + CName);
c.WL("Company Address is:-" +CAddress);
}
```

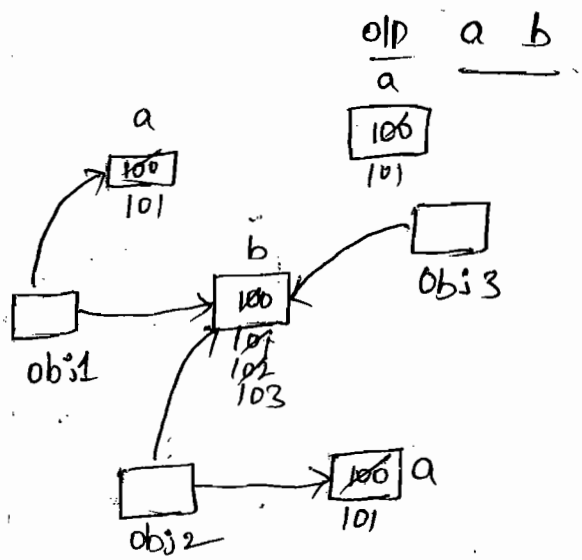
```
class Example6
{
  static void main ()
  {
    Employee6 obj1 = new Employee6();
    Employee6 obj2 = new Employee6();
      obj1.DispEmpData();
      obj2.DispEmpData();
  }
}
}
```

→ Another Example using Static Constructor.

Code:-

```
namespace CAConstructors
{
  class sample
  {
    int a; static int b;
    public Sample()
    {
      a = 100;
    }
    static Sample()
    {
      b = 100;
    }
    public void Display()
    {
      C.WL(a + "\t" + b);
      a++;
      b++;
    }
  }
```

Sample
int a static int b
Public Sample () Static sample () Public void Display()



```
class Example7
{
static void Main()
{
Sample obj1 = new sample();
Console.writeline("a\t\n----");
obj1.Display();
Sample obj2 = new sample();
obj2.Display();
sample obj3 = new sample();
obj3.Display();
Console.Read();
}
}
}
```

If we can call again "obj1.Display" after obj3, the o/p will be

a	b
100	100
100	101
100	102
101	103

o/p

a	b
100	100
100	101
100	102

*"Default accessability" of main is "private"

14/11/2014

Instance Constructors:-

These constructors will maintain separate instance of the data fields, for each object created to the class. So these are called instance constructors.

Non-Instance Constructors:-

These will maintain only one instance of these data fields for any no. of objects created to the class. so these are called non-instance constructors.

Why main method is static?

In general to consume the class, that contains main method we are n't creating any object, rather we are changing startup object class name, even from command prompt we are using class name so main method is n't being called using object name rather called using class name

Any member that we would like to calling using class name should be static. So we are n't calling main method using object name rather calling using class name. so main method is static.

Function Signature (or) Method Signature:-

In general, when we take any function or method, it contain's two parts.

(i). Signature
(ii). Implementation

(i) Signature of method:-

Method signature includes following

(a) Function Name or method Name

(b). Number of Arguments and datatypes of arguments.

But function Signature doesn't include "access modifier" and "Return type"

(ii) Function Implementation:-

Function implementation is nothing but the code that we write within the function.

Any two functions contain function name, no. of arguments and datatypes of arguments same. They are called with same signature otherwise they are called with different signature.

```
public void Add (int a, int b)
{
  code
  :
}
```

Add (int a, int b) → method signature/prototype/Declaration

code → method Body/Definition/Implementation

Function or method overloading:-

In general assigning some additional work to a function apart from the existing work, is known as function overloading, in general, a function will be overloaded in two senario's.

(i). when no. of arguments are changed.

(ii). when datatypes of arguments are changed.

(i). when no. of arguments are changed:-

If we consider a function like

```
public void Add(int a, int b)
{
  :
}
```

We can call these function by passing two integer values like

Add (20,30).

But if you would to like to call this function by passing ~~two~~ with different ~~type of values~~, like ~~Add(20, 30.5)~~ no. of arguments like

Add (10, 20,30)

rather we overload the function like ~~Add(20,30.5)~~

```
public void Add (int a, int b, int c)
{
 :
}
```

(ii) When datatype of arguments are changed:-

If we consider a function like public void ~~int~~ Add(int a , int b)

```
{
 :
}
```

we can call this function by passing two integer values like

Add (20,30)

But if you would like to call different type of values like Add(20,30.5). It is n't possible rather we create an another function like,

```
public void Add (int a , double b)
{
 :
}
```

***Technical Definition for function overloading:-

providing new implementation to a function, with ~~same~~ different signature but with same name, is known as function overloading

Function overloading can be implemented in same class or in Base and derived classes.

To implement function overloading, we don't use any separate keywords.

Function overloading is known as "code refinement technique".

15/11/2014

***Why Inheritance is Required:-

Let us assume that, a company has 'n' no. of branches where every branch is containing some no. of employees. company asked us to computarize branches details. Then we create a class like

Branch
int Bcode String BName String BAddress
public void GetBData() public void DispBData()

We created on this and gave to company. company is working very fine with this, later during the period of time, company also asked to computarized employee details of every branch.

so we understand the possible senario's for this purpose

Senario 1:-

we create a class Employee like

Employee
int BCode String BName String BAddress int EmpId String EName String Address int Age
public void GetBData() public void DispBData() public void GetEmpData() public void DispEmpData()

Here we created a new class Employee By writing again code for Branches Details.

Here we are unable to reuse that existing class code. If you are unable to reuse Existing class code there is no meaning of object oriented programming.

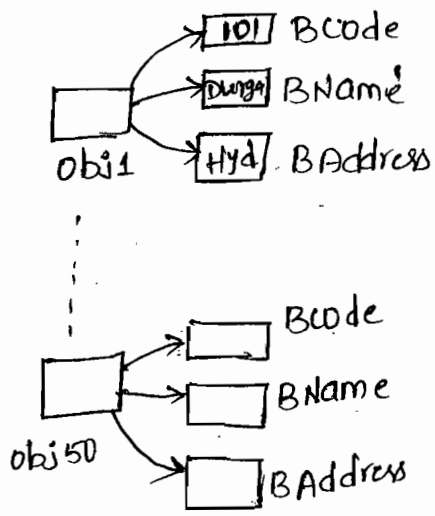
Senario 2:-

By creating Existing the class branch asusual, Let us create a new class employee like

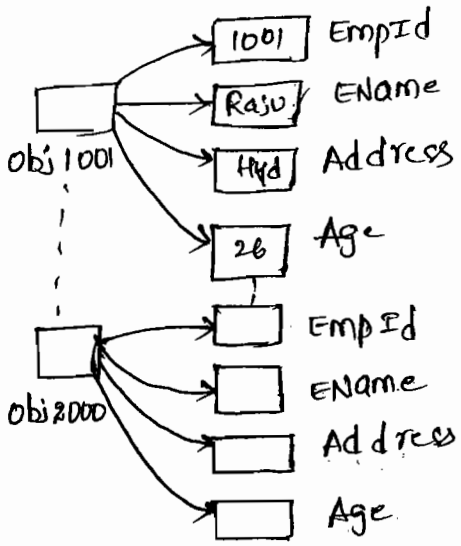
Branch
int Bcode String BName String BAddress public void GetBData() public void DispBData()

Employee
int EmpId String EName String Address int Age
Public void GetEmpData() public void DispEmpData()

Here we created branch class and Employee class separately, so to consume branch class, we create object like how many branches are there, that many objects like



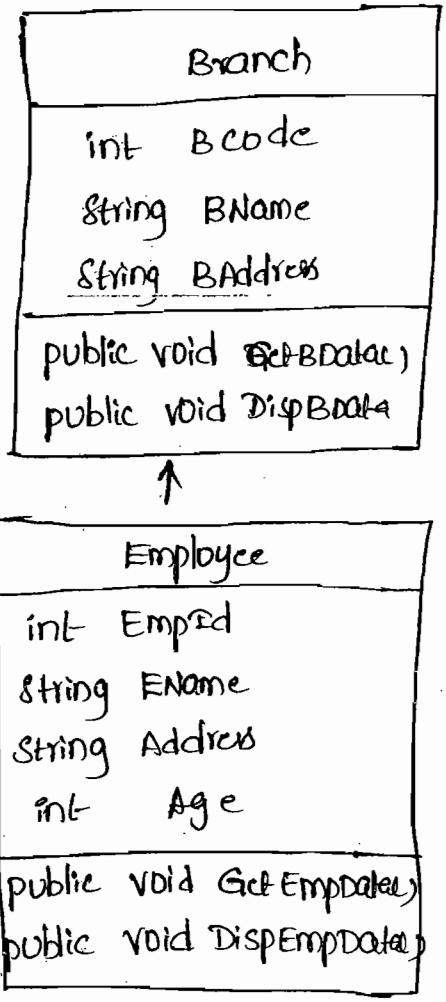
Similarly, How many Employees are there, that many objects created for employee class like



Here, though we are able to reuse the Existing class code, But the drawback in this is, "we are unable to identify which employee belongs to which Branch."

To overcome these drawbacks, we depend on inheritance. By using inheritance we create a new class "Employee" from Branch class like,

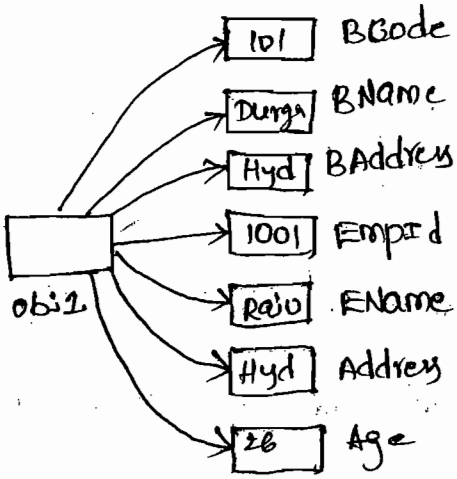
Here in inheritance process, derived class Employee will get, all the features of Base class branch.



To consume these two classes we create objects for the derived class employee like

Employee obj1 = new Employee();

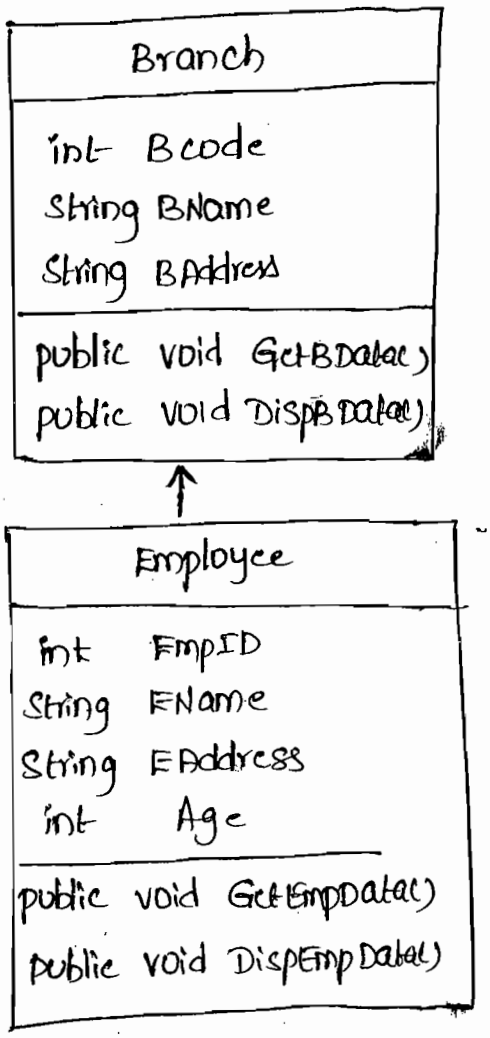
Then this object will represent, Both base and derived classes, like



By using inheritance, we are able to solve the drawbacks in both senario's.

We are able to reuse the existing class Branch code and we are able to identify which employee belongs to which Branch.

Example with ~~[illegible]~~ Single Inheritance:-



code:-

```
namespace CAInheritance
{
 class Branch
 {
  int Bcode; string BName, BAddress;
  public void GetBData()
  {
   Console.Write("Enter Branch Details:-");
   this.Bode = Convert.ToInt32(c.RL());
   this.BName = Console.ReadLine();
   this.BAddress = Console.ReadLine();
  }
  public void DispBData()
  {
   c.WL("Branch code is:-" +this.Bcode);
   c.WL("Branch Name is:-" +this.BName);
   c.WL("Branch Address is:-" +this.BAddress);
  }
}
~~class Emp~~ Em
```

```
class Employee : Branch
{
  int EmpId, Age; String EName, Address;
  public void GetEmpData()
  {
    Console.Write("Enter Employee Details:-");
    this.EmpId = Convert.ToInt32(Console.ReadLine());
    this.EName = Console.ReadLine();
    this.Address = Console.ReadLine();
    this.Age = Convert.ToInt32(Console.ReadLine());
  }
  public void DispEmpData()
  {
    Console.WriteLine("Employee Id is :-" + this.EmpId);
    Console.WriteLine("Employee Name is:-" + this.EName);
    Console.WriteLine("Employee Address is :-" + this.Address);
    Conside.WriteLine("Employee Age is :-" + this.Age);
  }
}
class Example1
{
  S v Main()
  {
    Employee1 Obj1 = new Employee();
      Obj1.GetBData();
      Obj1.GetEmpData();
      Obj1.DispBData()
      Obj1.DispEmpData()
  }  C.RL();
}
```

16/11/2017

* By using Inheritance we create a class for derived (Employee) class By using Base (Branch) class, and also we can create a class for Base class By using derived Class, By using an object " base".

" base" keyword:-

base is an object for all the base classes within the derived class. In the above Example by making GetBdata(), DispBData() functions as protected we can access from derived class. But n't from derived class. Then the code will be like

```
namespace CASInheritance
{
  class Branch
  {
    int Bcode; string BName, BAddress;
    protected void GetBData()
    {
      C.WL("Enter Branch Details:-");
      this.Bcode = C.ToInt32(C.RL());
      this.BName = C.RL();
      this.BAddress = C.RL();
    }
```

```
protected void DispBData();
{
    C.WL("Branch code is :-" + this.Bcode);
    C.WL("Branch Name is :-" + this.BName);
    C.WL("Branch Address is:-" + this.Address);
}
}
class Employee : Branch
{
int EmpId, Age; string EName, Address;
public void GetEmpData()
{
    base.GetBData();
    C.WL("Enter Employee Details:-");
    this.EmpId = C.ToInt32(C.RL());
    this.EName = C.RL();
    this.Address = C.RL();
    this.Age = C.ToInt32(C.RL());
}
public void DispEmpData();
{
    base.DispBData();
    C.WL("Employee Id is:-" + this.EmpId);
    C.WL("Employee Name is:-" + this.EName);
    C.WL("Employee Address is :-" + this.Address);
    C.WL("Employee Age is:-" + this.Age);
}
}
```

```
class Example1
{
  static void Main( )
  {
    Employee obj1 = new Employee();
      // obj1.GetBData();
      obj1.GetEmpData();
      // obj1.DispBData();
      obj1.DispEmpData();
      C.RL();
  }
```

Modifier's in C#:-

In C# modifier's are used to modify declaration of type and type members.

List of Modifiers in C#:-

1. abstract
2. async
3. const
4. event
5. Extern
6. new
7. override
8. partial
9. readonly
10. Sealed.
11. static
12. Unsafe
13. Virtual
14. Access Modifiers
    a. private
    b. protected
    c. internal
    d. protected internal
    e. public

Access Modifier's are used to set the accessibility levels of the class or interface or their memember's.

(a). private :- private member's are accessible within the same class and are n't accessible in any other class.

(b). protected :- protected member's are accessible within same class and also in derived class but are n't accessible in any other class. Irrespective to the derived class may be in same assembley or in any other assembly.

(c). internal :- Internal member's are accessible in any class within the same assembly. and are n't accessible in any class in any other ~~accessible's~~. assembly's.

(d). protected internal :- These are accessible in any class within the same assembly and also in derived class's in other assembly's. But are n't accessible in non-derived classes in other assembly's

(e). public :- public member's are accessible in any class and in any assembly.

Int. Q

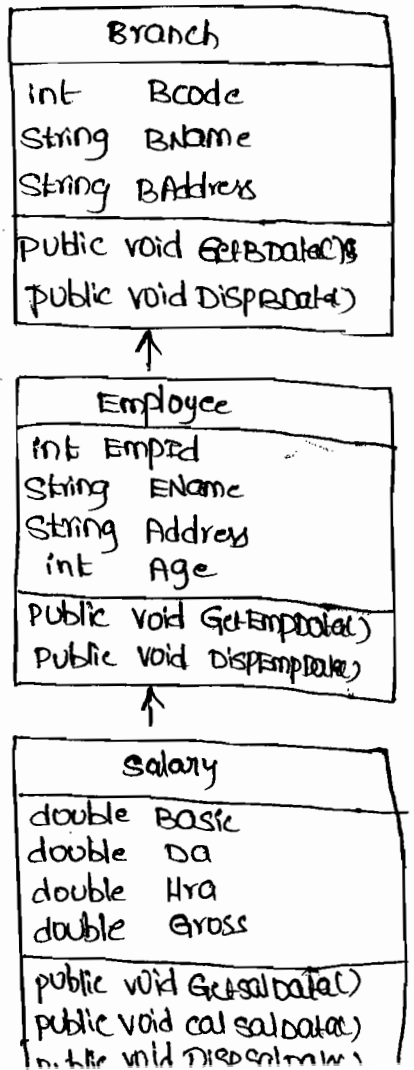
Access Modifier \ class	Same Assembly			Other Assembly	
	same class	Derived class	non-Derived class	Derived class	non-derived class
private	✓	✗	✗	✗	✗
protected	✓	✓	✗	✓	✗
internal	✓	✓	✓	✗	✗
protected internal	✓	✓	✓	✓	✗
public	✓	✓	✓	✓	✓

Multi Level Inheritance:-

Example with multilevel Inheritance:-

code:-

```
namespac CAInheritance.
{
class Branch
{
int Bcode; string BName, BAddress;
public void GetBData()
{
Console.Write ("Enter Branch Details:-")
this. Bcode = Convert.ToInt32(C.RL());
this. BName = C.RL();
this. BAddress = C.RL();
}
```



```
protected void DispBData();
public
{
  C.WL("Branch code is:-" +this.Bcode);
  C.WL("Branch Name is :-" +this.BName);
  C.WL("Branch Address is :-"+this.BAddress);
}
class Employee : Branch
{
  int EmpId, Age; string EName, Address;
  public void GetEmpData()
  {
    base.GetBData(); -> statement is optional
    C.WL("Enter EmployeeDetails:-");
    this.EmpId = Convert.ToInt32(C.RL());
    this.EName = C.RL();
    this.Address = C.RL();
    this.Age = Convert.ToInt32(C.RL());
  }
  public void DispEmpData();
  {
    base.DispBData(); --> statement is optional.
    C.WL("Enter Employee Id is:-" +this.EmpId);
    C.WL("Employee Name is:-" +this.EName);
    C.WL("Employee Address is:-"+this.Address);
    C.WL("Employee Age is:-"+this.Age);
  }
```

```
class ~~Employee~~ 6' salary : Employee ~~Branch~~.
{
  double Basic, Da, Hra, Gross;
  public void GetsalData()
  {
    C.WL("Enter Basic salary of employee:-");
    this.Basic = C.ToDDouble(C.RLL());
  }
  public void calculate()
  {
    this.Da = this.Basic * 0.4;
    this.Hra = this.Basic * 0.3)
    this.Gross = this.Da + this.Hra;
  }
  public void DispsalData()
  {
    C.WL("Basic salary is :-" + this.Basic);
    C.WL(" Da is :-" + this.Da);
    C.WL("Hra is :-" + this.Hra);
    C.WL("Gross salary is :-" + this.Gross);
  }
}
class Example1
{
  static void Main()
  {
    salary obj1 = new salary();
```

Function overriding:-

providing new implementation to a function with same signature, within the derived class, is known as function overriding.

To implement function overriding we use "virtual" keyword at "base level" and "override" keyword at "derived level".

Function overriding can be implemented in Base and derived class only and we can't implement within the same class.

Function overriding is known as code replacement Technique".

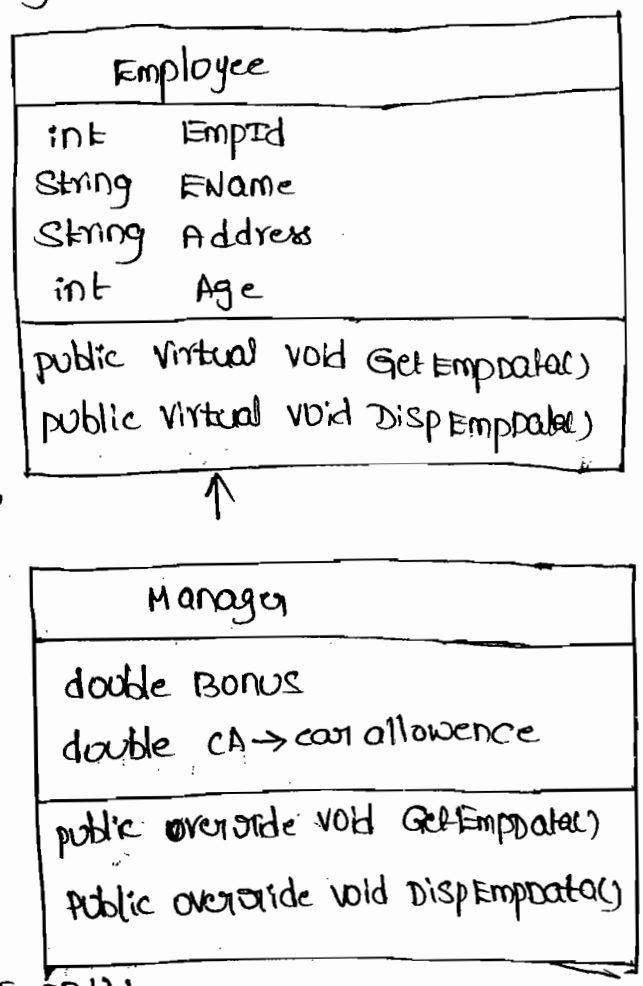
Function overriding is implement to dynamic polymorphism.

Differences B/W function Overloading & function overriding

S.NO	Function overloading	Function overriding
1.	providing New implementation to a function with Same Name and Different signature is known as function overloading	providing New implementation to a function with same Name and Same Signature is known as function overriding
2.	Function overloading will be done in the same class and also in the base and derived classes	Function overriding will be implemented in the Base and Derived classes. cannot implemented in Same class

3.	This is code Refinement technique	This is code Replacement Technique
4.	No separate keywords are used to implement function overloading	Use virtual keyword for base class function and override keyword in derived class function to implement function overriding
5.	used to implement static polymorphism	used to implement dynamic polymorphism.

Example with function overriding:-




```
namespace CAFOverride
{
 class Employee
 {
  int EmpId, Age; String EName,
                         Address;
  public virtual void GetEmpData()
  {
   c.WL("Enter Employee Details:-");

this.
this.
this.
this.
}
Public virtual void DispEmpData()
 {
  c.WL("Employee Id is:-"+ this. EmpId);

  c.WL("Employee Name is:-"+ this. EName);
```

```
C.WL("Employee Address is :-" +this.Address);
C.WL("Employee Age is :-" +this.Age);
public virtual void DispEmpDatal();
}
class Manager : Employee
{
    double Bonus, CA;
    public override void GetEmpDatal()
    {
        base.GetEmpDatal();
        C.WL(" Enter Manager Details :-");
        this.Bonus =: Convert.TODDouble (C.RLL());
        this.CA = Convert.TODDouble (C.RLL());
    }
    public override void DispEmpDatal()
    {
        base.DispEmpDatal();
        C.WL("Manager Id is :-" +this.EmpId);
        C.WL("Manager Name is :-" +this.EName);
        C.WL(" Manager Address is :-" +this.Address);
        C.WL(" Manager Age is :-" +this.Age);
        C.WL(" Manager Bonus is :-" +this.Bonus);
        C.WL(" Manager CA is :-" +this.CA);
    }
}
```

Intro Virtual Function:-

A function created by using virtual keyword, is known as virtual function.

→ Virtual function contains both signature and implementation.

→ overriding of a virtual function in derived class is optional.

→ If you don't override virtual function in derived class, it will call base class virtual function.

→ A virtual function can be written in a normal class or abstract class.

→ virtual function can't be written in sealed class or in interface

Abstract Functions and Abstract Classes:-

A function created by using abstract keyword is known as abstract function.

→ An abstract function contains signature only and doesn't contain any implementation.

→ An abstract function should be terminated.

→ override of an abstract function in derived class is mandatory.

→ If an abstract function is n't override in derived class then it raises compilation error.

→ An abstract function can be written within the abstract class or Interface

In a class following members can make as abstract.

(1), Functions

(2), Properties

(3), Indexers

(4), Nested Types

Note:- constructors (or) destructors can't be made as virtual or abstract

18/12/2014

Abstract class:-

A class that contains one or more abstract members is known as abstract class.

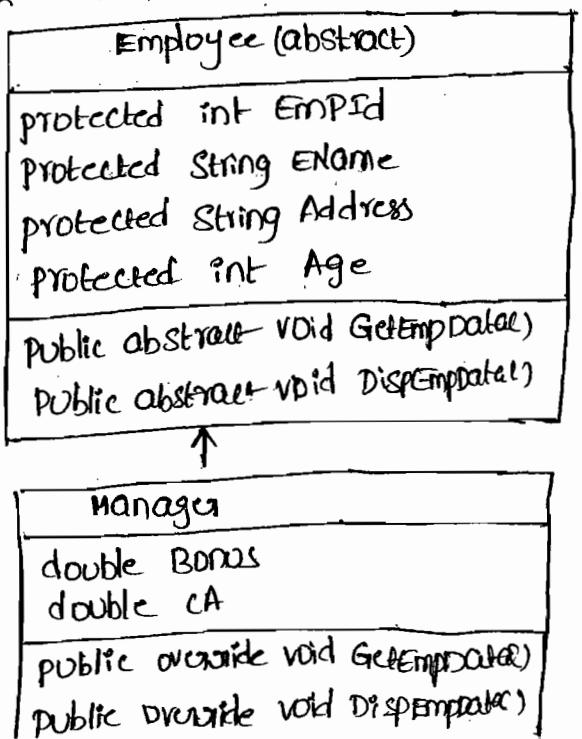
→ To create an abstract class use "abstract" keyword

→ An abstract class can't be instantiated directly, i.e, we can't object for abstract class

→ To consume an abstract class inherit a new class and create object for derived class.

Example with abstract class and abstract functions:-

```
namespace CAAbstract
{
  abstract class Employee
  {
  protected int EmpId, Age;
  protected String EName, Address;
  Public abstract void GetEmpData();
  Public abstract void DispEmpData();
  }
```



```
class manager : Employee
{
  double Bonus, CA;
  public override void GetEmpData()
  {
    C.WL("Enter Manager Details:-");
    base.EmpId = Convert.ToInt32(C.RL());
    base.EName = C.RL();
    base.Address = C.RL();
    base.Age = C.ToInt32(C.RL());
    this.Bonus = Convert.ToDouble(C.RL());
    this.CA = Convert.ToDouble(C.RL());

  public override void DispEmpData()
  {
    C.WL("Manager Id is:-" + base.EmpId);
    C.WL("Manager Name is:-" + base.EName);
    C.WL("Manager Address is:-" + base.Address);
    C.WL("Manager Age is:-" + base.Age);
    C.WL("Manager Bonus is:-" + this.Bonus);
    C.WL("Manager CA is:-" + this.CA);
  }
  class Examp1
  {
    Static void Main()
    {
      Manager obj1 = new manager();
        obj1.GetEmpData();
        obj1.DispEmpData())
        console.ReadLine();
    }
  }
```

Difference b/w virtual and abstract functions:-

S.NO	Virtual Function	Abstract Function
1	contains Both signature and implementation	contains signature only and doesn't contain any implementation
2.	use virtual keyword to create virtual function	use abstract keyword to create abstract function.
3.	overriding of a virtual function is optional	overriding of abstract function is compulsary
4.	If virtual function is n't overridden in derived class then base class virtual function is called	If an abstract function is not overridden in derived class then it raises compilation error.
5.	A virtual function can be written in abstract and non-abstract classes.	An abstract function can be written in an abstract class or in interface only
6.	A virtual function should not be terminated	An abstract function should be terminated

19/11/20

Working with Interfaces:-

An interface is a contract, between itself and the derived class. The contract is nothing but aggrement like interface will say that, it will provide all rules or specification's but no implementations Then derived class will agree, to implement all the rules ~~and~~ or specifications given by interface.

To create an interface we use, interface keyword. An interface can't contain following member's.

(1). Datafields
(2) Constructors
(3). Destructors
(4). Constants
(5). Non-abstract member's

An interface can contain all abstract member's only like

(1). Abstract functions
(2) Abstract properties
(3). Abstract indexers
(4). Abstract Events
(5). Abstract Nested Types
(6). Abstract operations

By Default interface member's are treated as public and abstract. So when creating interface member's don't use public and abstract.

By default run-time will treat that, interface member's are overridden in derived class. so when override interface member's don't use override keyword.

→ The main purpose of interfaces is, interfaces are used to provide control over the classes.

→ Interfaces can be used, to implement multiple inheritance.

→ An interface can't be instantiated directly.

→ From an interface we can create an interface or a class.

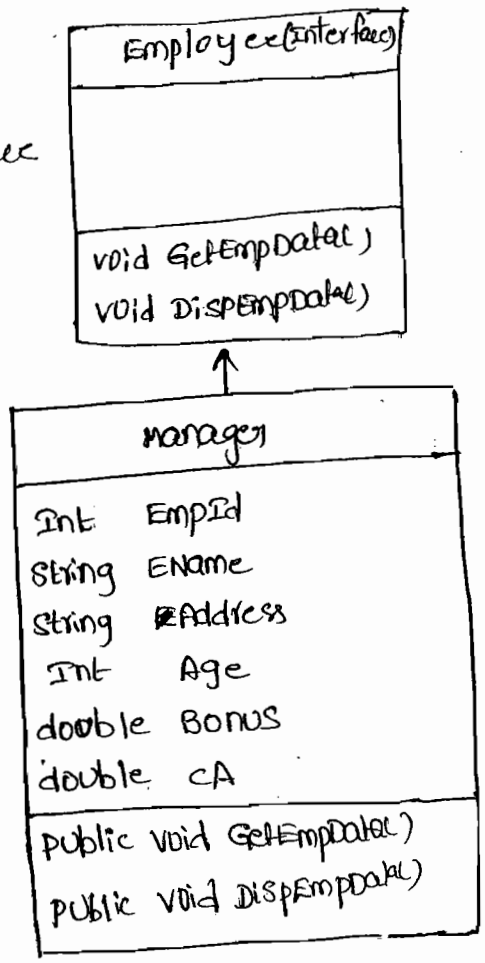
→ Any class that implement's the interface, should implement all the rules given by interface.

20/11/2014

Example with Interface:-

This is not original Example, just shown the example for interface

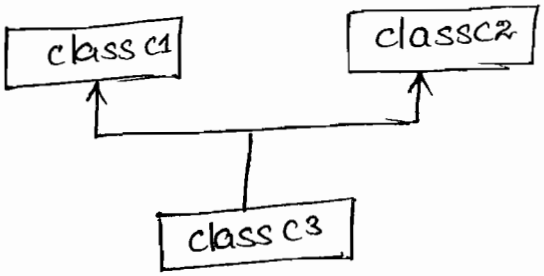
code:-
```
namespace CAInterface1
{
  interface Employee
  {
    void GetEmpData();
    void DispEmpData();
  }
  class manager : Employee
  int EmpId, Age;
  String Address, EName;
  double BONUS, CA;
```



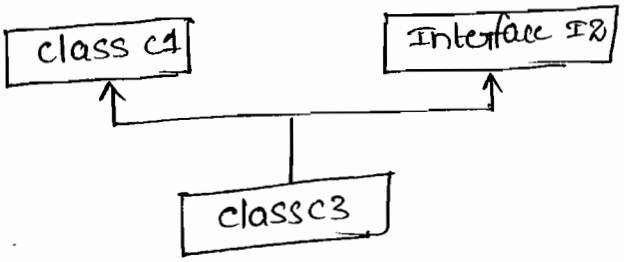
```
public void GetEmpData()
{
    C.Write("Enter manager Details:-");
    this.EmpId = Convert.ToInt32 (C.RL());
    this.EName = C.RL();
    this.Address = C.RL();
    this.Age = Convert.ToInt32(C.RL());
    this.Bonus = Convert.ToDouble (C.RL());
    this.CA = Convert.ToDouble (C.RL());
}
public void DispEmpData()
{
    C.WL("Manager Id is:- +this.EmpId);
    C.WL("Manager Name is:- "+this.EName);
    C.WL("Manager Address is :-"+this.Address);
    C.WL("Manager Age is:-" +this.Age);
    C.WL("Manager Bonus is:-" +this.Bonus);
    C.WL("Manager CA is:-" +this.CA);
}
class Example1
{
    Static void main()
    {
        Manager obj1 = new Manager();
        obj1.GetEmpData();
        obj1.DispEmpData();
            C.RL();
    }
}
}
```

Implementing multiple Inheritance in c#:-

To implement multiple inheritance in c#, there should be maximum 1 class at base level and minimum one interface at base level.

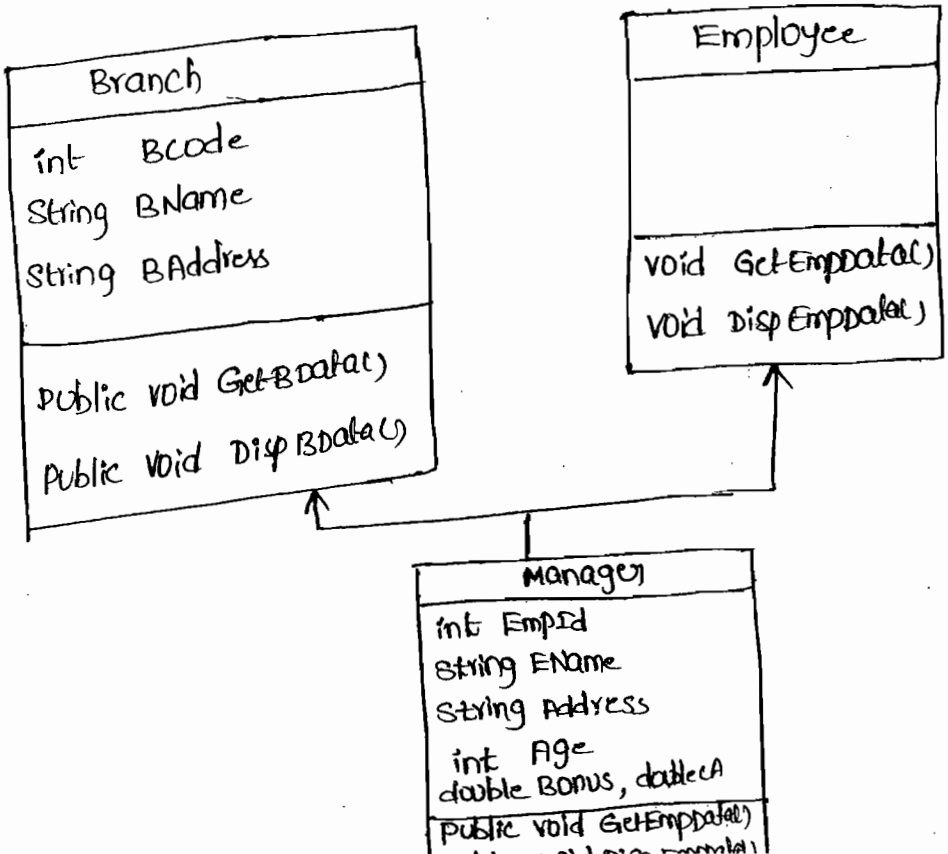


→ Not possible in c# and Java



→ Not possible in Java
possible in c#

Example to implement multiple inheritance in c#:-



code:-

```
namespace CA MPL Inhuritance.
{
  ~~interface~~ ~~Employee~~ class Branch
  {
    int Bcode, String BName, BAddress;
    {
     public void GetBData()
      {
        C.WL("Enter Branch details:-");
    this. Bcode = Convert.ToInt32(C.RL());
    this. BName = C.RL();
    this. BAddress = C.RL();
    }
    public void DispBData();
    {
    C.WL("Branch code is :-"+this. Bcode);
    C.WL("Branch name is :-"+this .BName);
    C.WL("Branch Address is:- "+this .BAddress);
    }
  class manager : Branch, Employee
  { int EmpId, Age; String EName, Address;
    double ~~CA~~. BONUS, CA;
    void GetEmpData();
    void DispEmpData();
    public void GetEmpData()
    {
      C.WL("Enter Employee Details:-");
      ~~C.WL("Employee Id is:- "+this.EmpId~~);
      this. EmpId = C.ToInt32(C.RL());
```

```
C.WL("Employee Name is :-"+this.EName);
C.WL("Empto
            this.EName = C.RL();
            this.Address = C.RL()
            this.Age = C.ToInt32(C.RL());
            this.Bonus = C.ToDouble(C.RL());
            this.CA = C.ToDouble(C.RL());

void DispEmpData()
{
  C.WL("Manager Id is:-"+this.EmpId);
  C.WL("Manager Name is:-"+this.EName);
   :
   :
  C.WL("Manager CA is:-"+this.CA);
}
class Example2
{ static void Main()
  { Manager obj1 = new Manager();
      obj1.GetEmpData();
      obj1.DispEmpData();
      C.RL();
  }
}
```

when we implement multiple inheritance we should use first class name followed by interface name like

```
class manager : Branch, Employee
{
   :
}
```

If you give first interface name followed by class name like

```
class manager : Employee, Branch
{
   :
```

It raises compilation Error like "Base class 'Branch' must come before any interfaces."

Why c# doesn't support multiple inheritance using 2 or more classes at base level :-

Ans: In general, if you make any rule in any language or mathematics or science. The rule should satisfy all the possible Senario's.

If the rule doesn't satisfy any single senario, then we say it as not a rule.

Senario1: If you consider two classes c1 and c2 like,

```
class C1
{ public void F1()
  {
  }
}
class C2
{
  public void F2()
  {
  }
}
```

If you create a new class c3, by inheriting from c1 and c2 like

```
class C3 : C1, C2
{
}
```

To consume these classes, we create object for c3 class and can call F1 and F2 functions like

```
c3 obj1 = new c3();
obj1.F1();
obj1.F2();
```

In this senario there is no problem F1 and F2 functions are called and executed comfortably.

Senario 2:-

If you consider two classes c1 and c2 like

```
class c1
{ public void F1()
  {
  }
}
class c2
{ public void F1()
  {
  }
}
```

If you create a new class c3 from c1 and c2 like

```
class c3 : c1, c2
{
}
```

To consume these classes, we create object for derived class c3 and call F1 like,

```
c3 obj1 = new c3();
obj1.F1();
```

Here, there is completely confusion to the run time it can't understand weather to call F1 function of c1 class or c2 class which makes ambiguity. Then this senario is not possible so we say c# doesn't support multiple inheritance using two or more classes at base level

How Interfaces solve this Problem:-

If you consider two interfaces like,

```
interface I1
{ void F1()
  {
  }
}
interface I2
{
  void F1()
  {
  }
}
```

If you create a new class from these interfaces like

```
class c3: I1, I2
{ public void F1()
  {
  }
}
```

to consume these interfaces and class we create object for C3 class and call F1 function like

```
C3 obj1 = new C3();
obj1.F1();
```

So Here, F1 functions of class C3 will call directly without any ambiguity.

Senario3:- If you consider one class and one interface I2 like, with same name like,

```
class C1
{
 public void F1()
  {
   ⋮
  }
}
interface I2
 {
  void F1();
   ⋮
 }
```

To create a class from the class C1 and interface I2 like

```
class C3 : C1, I2
{ public void F1()
  {
   ⋮
  }
}
```

To consume these class and interface, we create an object for C3 class and call function F1 like

```
C3 obj1 = new C3();
obj1.F1();
```

From scenario3, these implementation will be satisfied in C#, But not satisfied in Java.

21/11/2014

Example to implement multiple inheritance using two or more interfaces at base level :-

```
namespace CAInterface2
{ interface I1
  {
    void F1();
  }
  interface I2
  {
    void F1();
  }
  class C3: I1,I2
  {
    public void F1()
    {
      C.WL(" This is implementing F1() function of I1 and I2
              Interfaces");
    }
  }
class Example1
{ static void Main()
  {c3 obj1= new c3();
   obj1.F1();
   console.Read();
  }
}
}
```

Note:- In the above Example, there is a drawback that, F1 function of I1 and I2 interfaces is getting same implementation within the derived class C3. But in real time we may have separate implementation to the F1() function of I1 and I2 interfaces within the derived class C3.

For this purpose we depend on Explicit interface implementation.

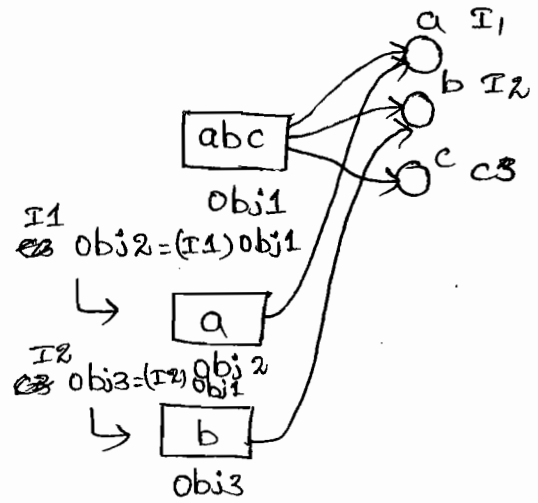
*** Explicit Interface Implementation:-

providing separate implementation, to the same function of two or more ~~functions~~ interfaces within the derived class is known as explicit interface implementation.

Example with Explicit Interface Implementation:-

```
namespace CAIntrfaces
{
  interface I1
  {
    void F1();
  }
  interface I2
  {
    void F1();
  }
  class C3: I1, I2
  {
    void I1.F1()
```

```
{ C.WL(" This is overriding F1() function of interface I1");

void I2.F2()

{ C.WL(" This is overriding F2() function of interface I2");

}
class Example1

{
static void main()
 {
   C3 obj1 = new C3();

   obj1.F1();

  I1 obj2 = (I1) obj1;

  I2 obj3 = (I2) obj1;

   Obj2.F1();
   obj3.F1();
   C.Read();
 }
}
}
```



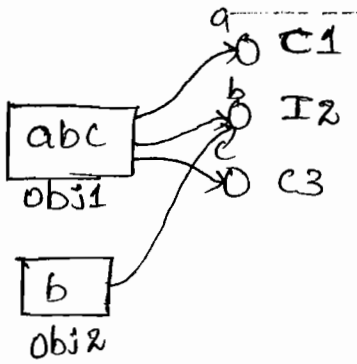
***

Note :- Java doesn't support explicit interface implementation.

A problem which is n't solved in Java :-

```
namespace CAInterface
{ class C1
  {public void F1()
   {C.WL("This is F1() function of C1 class");

   }
}
```

```
interface I2
{ void F1();
}
class C3 : C1, C2
{ void I2.F1()
  {
    C.WL("This is overriding F1() function of I2 interface");
  }
}
class Example1
{ static void Main()
  {
    C3 obj1 = new C3();
      obj1.F1();
    I2.obj2 = (I2)obj1;
      obj2.F1();
      C.Read();
  }
}
}
```



NO GAP

Int. Q) Differences between Abstract class and Interface:-

S.NO	Abstract class	Interface
01	A class which contain one or more abstract functions is known as abstract class	→ An interface is a contract between itself and the derived class
02	An abstract class can contain non abstract members	→ An interface cannot contain non abstract function members
03.	An abstract class can contain all members of class	→ An interface can contain abstract functions, properties, indexers, Events only and can't contain non-abstract functions, Data fields, constructors and destructors, constants, also.
04.	Use abstract and class keywords to create an abstract class	→ use interface keyword to create an interface
05.	An abstract class can't be used to implement multiple Inheritance	→ An Interface can be used to Implement multiple Inheritance
06.	By default abstract class members not public and not abstract	→ By default Interface members are public and abstract.
07.	An abstract class can't be instantiated directly	→ 07. An interface can't be instantiated directly
08.	Creating a new class from an abstract class is mandatory in order to consume it.	→ creating a new class from interface is mandatory in order to consume it.

Working with partial class:-

A class that will allow the user to divide the code into multiple files is known as partial class.

To create a partial class we use a keyword "partial"

*** → Java doesn't support partial classes.

Advantages using partial classes:-

1), Application can be developed at faster rate.

2), Enhancements can be made easily.

22/11/2014

Example with partial classes:-

create a new console Application with the name CAPartial and write the following code

```
namespace CAPartial
{
  partial class Employee
  {
    int EmpId, Age; string EName, Address;
    public void GetEmpData()
    {
      Console.WriteL("Enter Employee Details:-");
      this.EmpId = Convert.ToInt32(C.RLL());
      this.EName = C.RLL();
      this.Address = C.RLL();
      this.Age = Convert.ToInt32(C.RLL());
    }
```

```
class Example1
{
  Static void Main()
  {
    Employee obj1 = new Employee();
      obj1.GetEmpData();
      obj1.DispEmpData();
      Console.Read();
  }
}
}
```

create a new class file with the name employee.cs write the following Code

```
namespace CApartial
{
  partial class Employee
  {
    public void DispEmpData()
    {
      c.WL("Employee Id is:-" +this.EmpId);
      c.WL("Employee Name is:-"+this.EName);
      c.WL("Employee Address is:-"+this.Address);
      c.WL("Employee Age is :-"+this.Age);
    }
  }
}
```

Run the application and check it.

Working with properties:-

In general, if we would like to access the datafields of a class from outside the class, we make datafields as public.

But we make data fields as public, there are many chances to loose the security and from outside the class, we perform two operations on datafields in general.

1). Reading the data from the datafields

2). Writing the data into the datafields.

But to perform these operations, we need to make datafields as public.

Without making data fields as public, we would like to perform read and write operations on the datafields, we use properties.

A property is used, just to transfer the data. To perform this read, write operations on the datafields, property contains two methods called accessors.

1). Set accessor

2) get accessor.

1). Set accessor:-

Set accessor is used, to write the data into data field.

Set accessor contains a fixed and default variable i.e, value.

→ When we write the data into data fields of a class, by using property. Then set accessor is called automatically and data will be return into datafields.

Syntax and Example of set accessor:-

syntax :-
```
set
{
    DataFieldName = value;
}
```

Ex:-
```
set
{
    EmpId = value;
}
```

(2). get accessor :-

This accessor is used, to read the data from the datafield of a class.

When a property is called, reading the data get accessor will be called by default and which return's the data.

Syntax and Example of get accessor:-

Syntax:-
```
get
{
    return DatafieldName;
}
```

Ex:-
```
get
{
    return EmpId;
}
```

Types of properties:-

C# supports 3 types of properties.

(1). Read only property

(2). Write only property

(3). Read-Write property.

(1). Read only property:-

This property is just used, to read the data from the data field of a class. This property contain only one accesser i.e, get-accesser

Syntax and Example of Read only property:-

Syntax:- AccessModifier Datatype propertyName

```
{
  get
  {
    return Datafield Name;
  }
}
```

Ex:- public int P1

```
{
  get
  {
    return EmpId;
  }
}
```

* ~~Write only~~ what should be the data type of the property?

12) Write only property :-

This property is used, to write the data into datafield of a class. This property contains only one accesser ~~to~~ i.e., setaccessor.

Syntax and Example of write only property:-

Syntax:- AccessModifier DataType propertyName

```
{
  set
  {
    DatafieldName = value
  }
}
```

Ex:- public int P2

```
{
  set
  {
    EmpId = value;
  }
}
```

when a property is called writing the data, setaccessor is called by default and data that we supply is transfered into value variable

13) Read-write property:-

This property is used, to perform both read write operations.

→ This property contains both set and get accessors.

Syntax:- AccessModifier DataType PropertyName

```
{
  set
  {
    DataFieldName = value;
  }
  get
  {
    return DataFieldName;
  }
}
```

Ex:- Public int P3

```
{
  set
  {
    EmpId = value;
  }
  get
  {
    return EmpId;
  }
}
```

* How many accessors can a property contain?

Ans:-

* can a property accept arguments?

Ans:-

* what is the default accessibility of the accessor?

Ans:-

* How to provide Data abstraction using properties?

1. Explain how do you provide security to the Data fields using properties?
2. How do you validate the data using properties?
3. can a property be overridden?
4. How do you make a property as Read only by writing both the accessors?
5. How do you make a property as write only by writing both the accessors?

6. What happens when a property is called for writing the data?

7. What happens when a property is called for reading the data?

8. It is possible to write the code outside the accessor with in the property?

23/11/2014

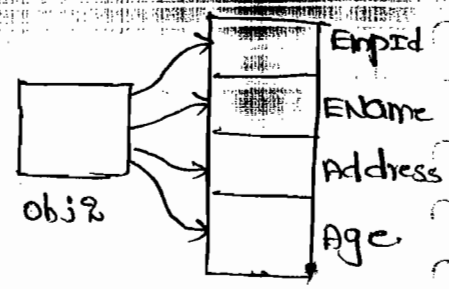
Example with read write properties?

[ie, providing Data abstraction using properties]

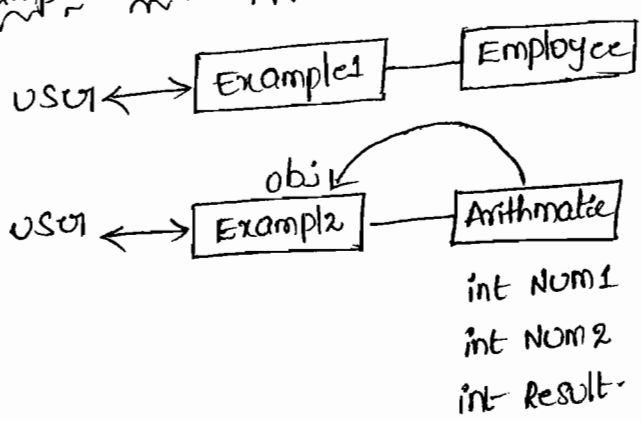
Employee	
int EmpId	
String EName	
String Address	
int Age	
public int PEmpId	RW
public String PEName	
public String PAddress	
public int PAge	

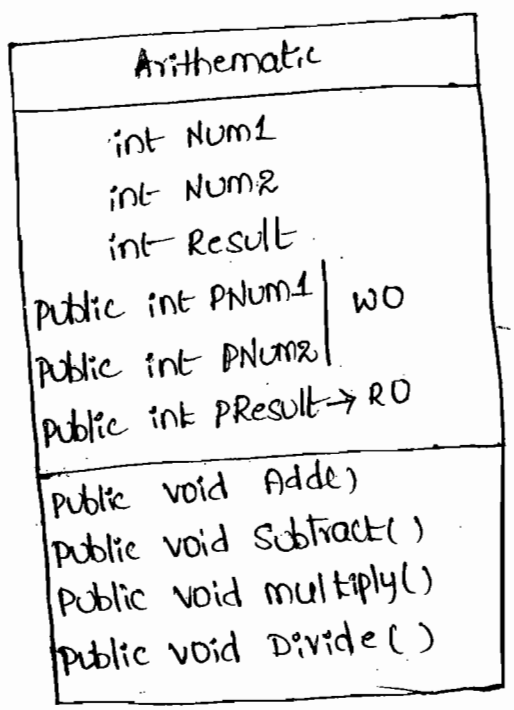
```
namespace CAproperties
{
  class Employee
  {
    int EmpId, Age; string EName, Address;
    public int PEmpId
    {
      set { EmpId = value; }
      get { return EmpId; }
    }
    public string PEName
    {
      set { EName = value; }
      get { return EName; }
    }
    public string PAddress
    {
      set { Address = value; }
      get { return Address; }
    }
    public int Age
    {
      set { Age = value; }
      get { return Age; }
    }
```

```
class Example1
{ static void main()
  {
    C.write("Enter Employee Details:-");
    Employee.obj1 = new Employee();
    obj1.PEmpId = C.ToInt32(C.RL());
    obj1.PEName = C.RL();
    obj1.PAddress = C.RL();
    obj1.PAge = convert.ToInt32(C.RL());
    C.WL("Employee Id is :-" + obj1.PEmpId);
    C.WL("Employee Name is :-" + obj1.PEName);
    C.WL("Employee Address is :-" + obj1.PAddress);
    C.WL("Employee Age is :-" + obj1.PAge);
    C.Read();
  }
}
}
```



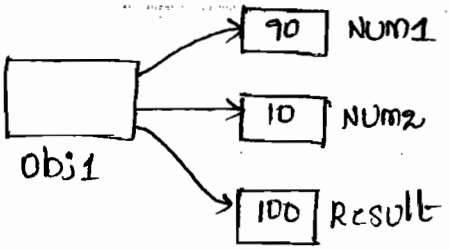
Example with Read only and Write only properties:-





code:-
```
namespace CAproperties
{ class Arithematic
  {
    int Num1, Num2, Result;
    public int PNum1
    {
      Set{Num1 = value;}
```

```
public int PNum2
 {
  set{ Num2 = value; }
 }
public int PResult
 {
  get{ return result; }
 }
public void Add()
 {
  Result = Num1 + Num2;
 }
public void Subtract()
 {
  Result = Num1 - Num2;
 }
public void multiply()
 {
  Result = Num1 * Num2;
 }
public void Divide()
 {
  Result = Num1 / Num2;
 }
}
class Example2
 {
  static void Main()
  {
   Arithematic obj1 = new Arithematic();
   C.W("Enter Any two Numbers:-");
   obj1.PNum1 = C.ToInt32(C.RL());
   obj2.PNum2 = C.ToInt32(C.RL());
   obj1.Add();
   C.WL("sum is :-" + obj1.PResult);
   obj1.Subtract();
   C.WL("Difference is:-" + obj1.PResult);
```



```
obj1.multiply();
C.WL("product is :-" + obj1.PResult);
obj1.Divide();
C.WL("Quotient is :-" + obj1.PResult);
C.Read();
}
}
}
```

In the above program, in object ~~class it contains~~ like

```
obj1.Add();
C.WL("sum is :-" + obj1.PResult);
obj1.PResult = 200;
```

This will ~~all~~ assign a value 200 to PResult, when in case of property will n't contain namespace. There is a chance to value allocated to PResult.

So for that we should always provide security to the data-fields by using properties. If properties are provided no chance to allocate unknown value "200".

Application:-

1. This type of security provided to the datafield in real time is in railway reservation site.

25/11/2014

~~For the above example~~ Example to use validations using properties:-

In the above Example if we would like to implement validation like num2 > value should not be grater than Num1 in that case num1 should equal to num2. Then we write the code like

code:- namespace CApropurties

```
{
 class Arithematic1
 {
  int Num1, Num2, Result;
  public int PNum1
  {
    :
  }
  public int PNum2
  { set
    { if (value > Num1)
          Num2 = Num1;
      else
          Num2 = value;
    }
  public int PResult
  { :
  }
  public void add()
  { Result = Num1 + Num2;
  }
    :
 }
 class Example3
 { Static void Main()
  { Arithematic obj1 = new Arithematic();
    e.w(
```

Symmetric and Asymmetric Accessors:-

When we write a property, By default accessability of the accessor is same as the accessibility of the property. i.e., If property is public set get are public, if property is private set get are private, if property is protected set get are protected.

Symmetric Accessors:-

If accessability of the both accessors is same, then they are called symmetric accessors.

```
public int PEmpId
{ set
  { EmpId = value;
  }
  get { return = EmpId;
  }
}
```

→ Here property is public, so second get are public. so both accessors are called symmetric.

Asymmetric Accessors:-

If accessability of both accessors is different, then they are called asymmetric accessors.

```
public int PEmpId
{ protected set
  { EmpId = value;
  }
  get
  { return EmpId;
  }
}
```

→ Here accessability of set is protected and get is public. so these are called asymmetric accessors.

I.Q** → where asymmetric accessors are used in real time?

Ans:- Asymmetric accessors are used in real time within the inheritance process more in real time.

Example of read only property by writing both accessors within the property:-

```
public int PEmpId
{
  private set
  {
    EmpId = value;
  }
  get { return EmpId;
      }
}
```

Example of write only property by writing both accessors within the property:-

```
public int PEmpId
{ set { EmpId = value;
      }
  private get
  { return EmpId;
  }
```

Auto Implemented properties:-

When we use auto implemented properties, the class doesn't contain any datafields. i.e., programmer doesn't create any datafields within the class.

Rather runtime will create the data fields in the class and data will read from these data fields and will be return into these data fields

→ we can't access the datafields from coding.

→ Auto Implemented property should contain both accessors and shouldn't have any implementation for set get

Example of Auto Implemented property:-

```
public int PEmpId
{
    set;
    get;
}
```

Example of Auto Implemented Readonly property:-

```
public int PEmpId
{ private set;
  get;
}
```

Example of Auto Implemented Writeonly property:-

```
public int PEmpId
{ set;
  private get;
}
```

→ Auto Implemented properties are introduced in 3.0 version of .NET framework.

***→ Auto Implemented properties are used in real time in OR (Object Relational) mapping tools.

***→ Advantages using Auto Implemented properties:-

Auto implemented properties are used to create light-weight classes. i.e., classes that will occupy less memory.

Example with Auto Implemented properties:—

```
namespace CAproperties
{ class Employee1
  {public int PEmpId
   { set;
     get;
   }
  Public string PEName
   { set;
     get;
   }
  public String PAddress
   { set;
     get;
   }
  public int Age
   { set;
     get;
   }
class Example4
{ static void Main()
  {Employee1 obj1 = new Employee();
   C.WL("Enter Employee Details:-");
   obj1.PEmpId = C.ToInt32(C.RL());
   obj1.PEName = C.RL();
   obj1.PAddress = C.RL();
   obj1.PAge = C.ToInt32(C.RL());
   C.WL("Employee Id is:-" +obj1.PEmpId);
   C.WL("Employee Name is:-" +obj1.PEName);
   C.WL("Employee Address is:-"+obj1.PAddress);
   C.WL("Employee Age is :-"+obj1.PAge);
  } C.Read();
```

Vs.Net	framework
✗	✓
✓	✗
8GB	500MB
with MSDN → 4GB	free HD.
Not free HD	

base.GetEmpDetail();

manager obj1 = new manager();

obj1.GetEmpDetail()

obj1.DispEmpDetail();

Express Edition → Free

professional → 7000/— /one copy

Ultimate → $12000 → /one copy

TFS (Team foundation Server) → $1Lakh → /one copy (60,lakh)

ALM Tool

All kind of features like

Para features

① MSIL code

② Ngen.exe

③ Native code

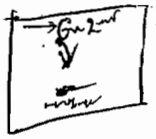
④ All the above

C.A.W.T.C.G

- .CS
- .CAS
- .C#
- .V#

C#

www.changeurjob.in



rajendra.cop@gmail.com

3G 2S

123124

13/12/2014

MULTITHREADING

Thread:- A Thread is a small piece of program.

i.e., when a program is running, the program is divided into slices. where each slice is called a thread.

* Threading is the concept of operating system, but n't concept of any programming language.

MultiThreading:-

Executing two or more such kind of threads is known as multithreading.

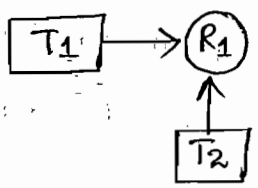
when we work with multithreading in general we face 2 problems.

(i) Illegal Cross Thread Calls

(ii) DeadLock.

(i) Illegal cross Thread Calls:-

When on thread is using a resource, at the same time other thread sends request for the same source then it is called illegal cross Thread calls.



To overcome this problem, we should keep the threads in ideal mode, allot the resource to other thread and swap this process until completion of thread execution.

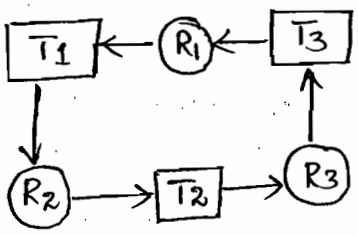
NO GAP ?

(ii) DeadLock :-

In a group of threads, if every thread is using some resource and waiting for other resource to leave the current resource. Then this situation will lead to dead lock occurence.

If a dead lock is occured, it leaves to Starvation (stopping of program), program Execution will n't completed further.

To overcome this we can use various kinds of predefined algorithms.



Here thread T1 is using resource R1 and waiting for resource R2. i.e., If it gets R2 resource then it releases resource R1. At the same time thread T2 is using resource R2 and waiting for resource R3. i.e., If it gets the resource R3 then it ready to release R2. At the same time thread T3 is using resource R3 and waiting for the resource R1. i.e., Thread T3 will release the resource R3 when it gets resource R1. so no thread is ready to release resource and will get resource. This is called DeadLock occurence.

To work with Threading in C# we have a dedicated Namespace System.Threading. This Namespace contains a class Thread.

Thread Class:-

(1). Non-static methods with thread class:-

(i). Abort( )

(ii) Resume( )

(iii). Start( )

(iv). suspend( )

(i). Abort( ):-

This method is used, to stop the thread Execution permanently.

(ii). Resume( ):-

This method is used, to stop the thread Execution of a suspended thread. The thread Execution will begin from where it was stopped.

(iii). Start( ):-

This method is used, to start or begin the thread Execution.

(iv). Suspend( ):-

This method is used, to stop the thread Execution temporarly. A suspended thread can be restarted using Resume method.

(2). Static methods with Thread class:-

(i). Sleep(Time in MS)

(i). sleep(Time in MS):- This method is used, to stop the thread Execution temporarly for some period, of time. After completion of the given time period, thread Execution will start Automatically.

Properties with Thread class:-

(i) IsAlive

(ii). Name

(iii). priority

(iv) Thread State.

(i) IsAlive :-

This property will return true, if the thread is n't stopped. otherwise will return false if the thread Execution is stopped.

(ii) Name :- used to set or get Name for the thread.

(iii) priority :-

used to set or get, the priority level for the thread.

All priority levels are avialable with an enumeration thread priority like

ThreadPriority :
- Lowest
- BelowNormal
- Normal (Default)
- AboveNormal
- Highest

(iv) → Lowest priority thread will gives less time for the Execution and highest priority thread will give more time for execution.

(iv) ThreadState :-

The priority will return the current state of the thread.

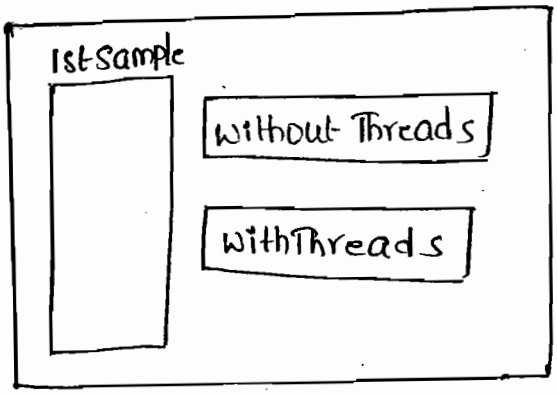
All thread states are avialable with an enumeration ThreadState like

ThreadState :
- Aborted
- BackGround
- Running
- stopped
- suspended.

***

Note : No two threads can run simultaneously

Example for Multithreading:-




```
Using System.Threading;

Thread T1, T2;

Private void F1()
{
  for(int i=1; i<=500; i++)
    1stsample.Items.Add(i);
}

Private void F2()
{
  for(int j=2001; j<=2500; j++)
    1stsample.Items.Add(j);
}

Private void btnWithoutThreads_Click(----)
{
  1stsample.Items.clear();
  F1();
  F2();
}

Private void btnWithThreads_Click(-----)
{
  1stsample.Items.clear();
  T1 = new Thread(F1);
  T2 = new Thread(F2);
  T1.Start();
  T2.Start();
}
```

In the above program there is possibility of illegal cross thread calls because the resource 1st sample is being accessed from two threads T1 & T2

To overcome this problem write the following statement in the Form_Load Event.

```
private void Form1_Load(-----)
{ control.CheckForIllegalCrossThreadCalls=false;
}
}
```

WINDOWS USER CONTROLS

Working with windows user controls:-

A user control in windows forms is a reusable component. Windows user controls are in the form of dll's in general. To create these user controls in windows forms we use windows forms and control library, template. once a windows user control is created it can be added to tool box and can be reused for any number of items.

Advantages Using windows User Controls:-

(i). Reusability

(ii). Less burden on the programmer

(iii). will provide RAD (Rapid Application Development) Facilities.

In real time projects first windows user controls will be created and then only windows forms will be created. To work with windows user controls we need to use following steps.

Step 1 : create user control

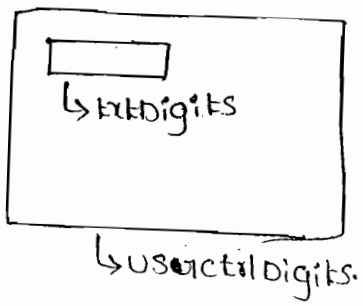
Step 2 : write the required logic code.

Step 3 : Build and test the user control

Step 4 : consume the user control.

Example to working with user controls:-

create textbox that accepts Digits, only



Step 1:- create windows user ctrl.

→ open visual studio.Net

→ click on File, click on New

→ click on project

→ select windows installed templates and select class library template

→ Type the application Name with in paranthesis" WinCLibDigits"

→ click on ok.

→ chang the name of the user control to UserCtrlDigits.

→ create textbox

→ change the following properties of textbox

→ ~~create textbox~~ Name → txtDigits

→ Dock property → Fill

→ Multiline to True and modifier property to public

Step2: write the required logic code

→ Goto keypress event of textbox txtDigits and write the following code

```
private void txtDigits_keypress(----)
  if(!char.IsDigit(e.KeyChar)
```

Step3: Build and test the user control.

→ click on build

→ click on build solution (s.c. key ctrl+F6)

This will create dll, run the application to test it.

It will open in a separate test container, enter digits, non-digits and check it.

Step 4:- consuming the user control:-

→ create new windows forms application.

→ Goto toolbox, click with RM Button,

→ click on Add tab and type tab name as usercontrols.

→ click with RM Button

→ click on choose items the select .Net Framework Components.

→ click on browse, goto location where dll is saved.

→ select the dll WinCLibDigits.dll

→ click on open, click on ok.

This will add UsrCtrlDigits control to the toolbox, Double click on this control, which create instance for this controls.

WORKING WITH PROCESS CLASS

process class is avialable in "system.Diagnostics" Namespace which contains various methods and properties, which are helpful to create and manage the process, running under the control of operating system.

Properties with Process class:-

1. ExitTime
2. HasExited
3. Id
4. MachineName.
5. StartInfo
6. StartTime
7. UserProcessorTime
8. TotalProcessorTime
9. ProcessName.

1. ExitTime:-

This property is used to set or get time when the process should be stopped.

2. HasExited:-

This property will return true, if the process Execution is stopped. otherwise it will return false.

3. Id:-

This property will return the Id that is associated with the current process.

4. MachineName:-

This property will return name of the machine in which process is created.

5. StartInfo:-

This property will return the information that is associated with the process when process is started.

⑥ StartTime :-

This property is used to set or get the time when process should begin.

⑦ UserProcessorTime:-

This property will return the time period that processor has spent to run the process without ideal time.

⑧. TotalProcessorTime:-

This property will return the total time spent by the processor including ideal time.

⑨. ProcessName:-

This property is used to set or get name for the process.

Non-Static Methods with process class:-

① close()
② kill()
③ Start()

①close():-

This method is used to free, all the resources associated with the process, so that the resources can be used for other purpose.

②, kill():-

This method is used to stop the process immediately.

③ Start():-

This method is used to start the process that is associated with the process class object.

Static methods with process class:-

① GetCurrentProcess ()

② GetProcessById (int Id)

③ GetProcessByName (String Name)

④ GetProcess ();

⑤ Start (String Name)

①. GetCurrentProcess ():-

This method is used to get the process that is associated with current application.

②. GetProcessById (int Id):-

This method will return, the process when its Id is given.

③. GetProcessByName (String Name):-

This method will return the process when its name is given.

→ Return type of above 3 methods is "Processclassobject".

④. GetProcess ():-

This method will return the ~~process~~ the list of all processes running under the control of operating system in current machine. If machine name is supplied then it returns list of the processes running in that m/c.

→ Return type of this method is "ProcessclassArray"

Ex:- Process [ ] P = Process. GetProcess ("computer");

⑤. Start (String Name):-

This method is used to create a new process for the given file

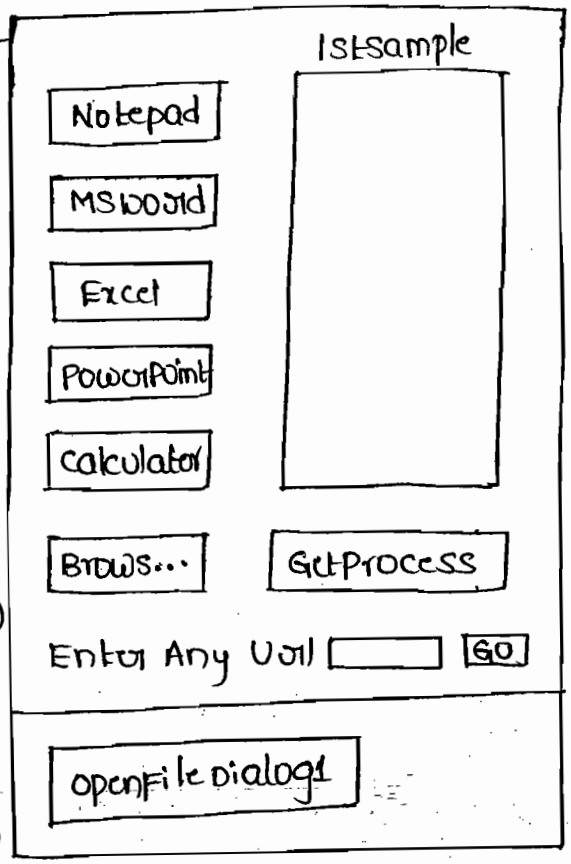
Ex:- for Notepad files → Process. Start ("Notepad. Exe");

for Normal files → process. Start ("D:\\sample\doc");

for URL → process. Start ("http:\\www. yahoo. com");

14/12/2017

1. Example to work with Process class:-




```
using System.Diagnostics;
namespace WAProcess.
{
 Public partial class Form1:Form
 {
  Public Form1()------
 private void Form_Load(----)
 {
 }
 Private void btnNotePad_click(---)
 {
  Process.Start("Notepad.Exe");
 }
private void btnMSWord_click(---)
{
 Process.Start("winword.Exe");
}
private void btnExcel_click(----)
 {
  Process.Start("Excel.Exe");
 }
Private void btnPowerPoint_click(----)
 {
  Process.Start("powerPnt.Exe");
 }
Private void btncalculator_click(----)
 {
  Process.Start("calc.Exe");
 }
 Private void btnBrowse_click(-----)
 {
  openFileDiaglog1.ShowDialog();
  process.Start(openFileDialog1.FileName);
 }
```

```
private void btnGO_click(----)
{ process.Start("txtURL.Text");
}
Private void btnGet_click(----)
{ Process[ ] P = Process.GetProcess();
  foreach(Process x in P)
   lstsample.Items.Add (x.ProcessName);
 }
}
}
```

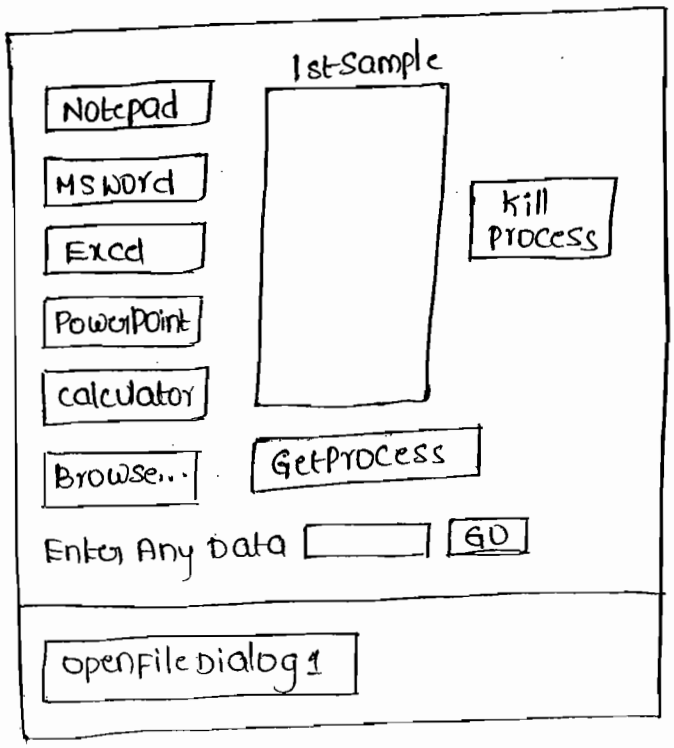
H.W

Example2:

In example1 there was a drawback avialable i.e, First time we click on get process all the process will be displayed in lstsample, Again we click on GetProcess second time all the process displayed or repeated, Again we click on GetProcess third time all the Process will be displayed, on lstsample, and soon --- so how can you rectify this problem.

H.W

Example3:



WINDOWS SERVICES

Working with windows services:-

windows services are the components that run under the control of operating system. These services will help the other applications to run, also support the operating system activities.

Every service has following start types,

(i), Automatic

(ii), manual

(iii), Disabled.

(i), Automatic :-

These services will be started, when operating system is booted.

(ii), Manual :-

These services will n't be started when operating system is booted rather user need to start manually or some other program to start.

(iii), Disabled:-

These services will be started, when they are enabled.

Every windows service has following events associated with it.

(i), on start()

(ii) onstop()

(iii) onpause()

(iv) onresume()

(i), onstart() :-

This Event will be fired, when a windows service is started at a control panel.

(ii) onstop():-

This event will be fired when a windows service is stopped at control panel service.

(iii) onpause():-

This Event will be fired, when a windows service is paused, at a control panel.

(iv) onResume():-

This Event will be fired, when a paused service is restarted at control panel service.

To create a windows service we use windows service Template in visual studio.Net.

To work with windows service we should use the following steps.

(i). create the service

(ii). write the required Logic code

(iii). Set the properties

(iv). Build the Service.

(v). place the service in control panel Services.

Example to create a windows service to record System StartTime and ShutdownTime:-

Step (i) creating the Service:-

→ Goto visual Studio.Net

→ click on File, click on New, click on project.

→ Select windows from visual c# from installed templates.

→ select windows service template then type the service name (LoginRecord), choose location to save, click on ok.

→ click on the link, click here to switch to code view

→ Goto solution explorer, change the service class name from Service1.cs to login.cs.

Step(ii):- writing the required logic code:-

→ Goto login1.cs file, write the following code.

```
using System.IO;
namespace LoginRecord
{ public partial class Login: ServiceBase
  {
   public Login()------
   protected override void onstart(string[ ] args)
   {
    StreamWriter SW = new StreamWriter("D:\\sample\\sample1.txt",
                                                        true);
    SW.WriteLine("LoginTime is:-" + DateTime.Now);
    SW.Flush();
    SW.close();
   protected override void onstop()
   {
   StreamWriter SW = new StreamWriter("D:\\sample\\sample1.txt",
                                                        true);
   SW.WriteLine("Logout Time is:-" +DateTime.Now);
   SW.Flush();
   SW.close();
  }
```

Step (iii):- Set the required properties:-

→ Goto login.cs Design

→ click with RM Button, click on Add installer, this will create two components.

(i), ServiceInstaller1

(ii), ServiceProcessInstaller1.

→ select service Installer1 and set the following properties.

(i), DisplayName → LoginService

(ii), ServiceName → Login

(iii), StartType → Manual

→ Then select the ServiceProcessInstaller1 and set the following properties.

(i), Account → Local installer.

Step(iv): Build the Service:-

→ click on Build, Build Solution (F6)

→ This will create an Exe with the name LoginRecord.Exe

Step(v): Placing the service in control panel Service:-

→ Goto visual studio command Prompt

→ chang the disk location to Exe location.

→ Type the following.

installUtill.Exe -i LoginRecord.Exe

then you get the following message "The commit phase completed successfully.

The transacted install has completed.

→ Goto control Panel, double click on Administrative tools, double click on services

→ select Login Service, click with RM Button, click on Start

→ Goto D: Sample Folder → open sample.exe, then you find the Login Recorded time.

→ change the service start type to Automatic, restart the computer and check it.

27/12/2014

WORKING WITH MDI APPLICATIONS
[MULTIPLE DOCUMENT INTERFACE APPLICATIONS]

In general GUI (Graphical User Interface) environment, we find two types of applications.

(1). SDI Applications

(2). MDI Applications.

(1). SDI (Single Document Interface) Applications:-

The application that will allow the user to work with only one file or only one document at once are known as SDI applications.

Examples of SDI applications:-

Notepad, WordPad, PaintBrush, etc....

(2) MDI (Multiple Document Interface) Applications:-

The applications that will allow the user to work with more than one file or more than one document at once are known as MDI applications.

Examples of MDI Applications:-

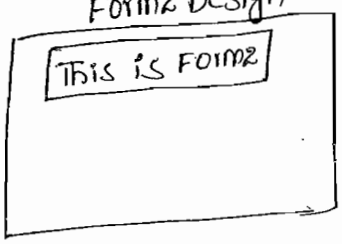
Application Structure :-

WAMDI
- → Form1
- → Form2
- } → child Forms
- → FormMDI → parent Forms.

Design Forms:-

Form1 Design

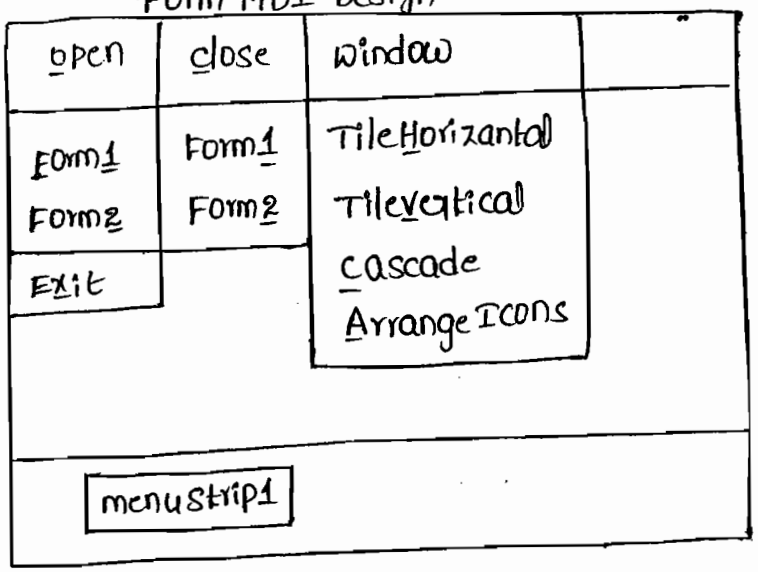
Form2 Design



Example with MDI application:-

To get menustrip:
common controls
↓
menustrip.

FORM MDI Design



→ create a new windows form application with the name WAMDI

→ create Forms Form1, Form2, FormMDI and Design

→ set the following property's for form MDI.

IsMDIContainer → True

Windows State → maximized.

→ In Form MDI write the following code

```
{ Form1 obj1 = new Form1();
  Form2 obj2 = new Form2();

  Public FormMDI( )...

  private void SmnuOForm1_Click(----)
  { obj1.mdiParent = this;
    obj1.Show();
  }
  private void SmnuOForm2_Click(---)
  { obj2.mdiParent = this;
    obj2.Show();
  }
```

```
private void smnuExit_click(---)
{ Application.Exit();
}
private void smnuCForm1_click(---)
{ obj1.Hide();
}
private void smnuCForm2_click(---)
{ obj2.Hide();
}
private void smnuHorizantal_click(---)
{ this.LayoutMdi(MdiLayout.TileHorizantal);
}
private void smnuVertical_click(---)
{ this.LayoutMdi(MdiLayout.TileVertical);
}
private void smnucascade_click(---)
{ this.LayoutMdi(MdiLayout.cascade);
}
private void smnuArrangeIcons_click(---)
{ this.LayoutMdi(MdiLayout.ArrangeIcons);
}
```

Exercise:-

If you click on Form1, Form1 is opened and click on Form2, Form2 will be opened. If you click on Exit two Forms (Form1 and Form2) are closed. Again you click on Form1 and Form2 in open File ~~Dialog~~ Button, Form1 and Form2 will be opened. upto know it's

working very fine. If Now if you click on close symbol [X] on rightside top corner of form1 and Form2. Then two Forms (Form1 and Form2) will be closed. Again goto openFile button and click on Form1 and Form2, then Forms will n't be opened. So what is the reason n't opened Form1 and Form2 and how to rectify this and write the code.

where MDI forms are used in realtime:-

In real time, first modules are developed in projects, then all the modules are integrated together by using the MDI forms.

03/01/2015

XML ASP.NET

→ We know that, Internet has been evolved by the year of 1990's. Every company started using internet, But when every company started using internet then new Problem will be raised. i.e., communicating among different companies developed has been developed because every company follow it's own proprietary format and one technology format is not understood by other technology.

→ This made companies to search for a new technology and w3org proposed to make some open standard technology then XML has been evolved from this idea.

→ XML is in plain text format and any technolgy can understand XML format.

→ XML contains only one version. i.e., 1.0.

→ XML is a tag based Programming language.

→ XML contains only one root tag. Each root tag contains one or more parent tags. Each parent tag contains one or more child tags.

→ In general parent tag represents database name, parent tag represents table name, child tag represents field name

→ Every XML file begins with start document like.

<? XML version = "1.0" encoding = "utf-8" ?>

→ structure of XML file is.

```
<Root tag>
  <parent tag1>
    <childTag1> Data </childTag1>
    <childTag2> Data </childTag2>
  </ParentTag1>
  <ParentTag2>
    <childTag1> Data </childTag1>
    <childTag2> Data </childTag2>
  </ParentTag2>
</Root Tag>
```

→ XML code can be written in any text editor.

→ XML file will have a default. Extension of ".xml"

→ xml code can be run in any browser. Every browser will contain xml Parser, to parse and run the xml code.

→ VS.Net will containg building editor and Parser facilities.

→ most populary used encoding format in xml is UTF8. developed by "Ken Thomson."

UTF means universal Transmission Format

In general in xml we find 3 types of files.

(1), xml : To describe the data, & store the data.

(2), xsd : xml Schema document, To store the schemas.

(3), XSLT : To store the style scheets

Creating an Xml File.

If we have two tables like

EmpDetails

EmpId	EName	Designation	DOJ	salary	DeptNO
101	Raju	Programmer	01/01/2014	15000	10
102	Abhi	DSE	02/02/2014	10000	20
103	Surya	TL	03/03/2014	35000	10
104	Teja	SysProgrammer	04/04/2014	35000	20
105	Sai	PL	05/05/2014	80000	10.

Dept

Deptno	Dname	Location
10	coding	Hyd
20	Networks	Banglore
30	Database	Mumbai

→ Take a notepad application, write the following code.

```
<?xml version="1.0" encoding="Utf-8"?>
<Employee>
  <EmpDetails>
    <EmpId> 101 <|EmpId>
    <EName> Raju <|EName
    <Designation> Programmer <|Designation>
    <DOJ> 01/01/2014 </DOJ>
    <salary> 15000 <|salary>
    <Deptno> 10 <|Deptno>
  <|EmpDetails>
  <EmpDetail>
    <EmpId> 102 <|EmpId>
    <EName> Abhi <|EName
```

all are in lower case only.

```
    <Designation> DSE </Designation>
    <DOJ> 02/02/2014 </DOJ>
    <Salary> 10000 </Salary>
    <Deptno> 20 </Deptno>
    !
    !
    !
  <Dept>
    <Deptno>10 </Deptno>
    <Dname> coding </Dname>
    <Location> Hyd </Location>
  </Dept>
  <Dept>
    <Deptno> 20 </Deptno>
    <Dname> networks </Dname>
    <Location> Banglore </Location>
  </Dept>
  <Dept>
    <Deptno> 30 </Deptno>
    <Dname> Database </Dname>
    <Location> mumbai </Location>
  </Dept>
</Employee>
```

→ save the file as sample.xml, open the file in browser and check the result.

→ you find the result with expandable and collpsable symbols.

Creating an xml file in VS.Net:-

→ create new windows forms application with the name WAXml

→ Goto solution Explorer, select the solution

→ click with RH Button, click on Add, click on New Item.

→ select xml file template, type the xml file name (sample.xml)

→ click on Add, write the following code.

```
<? version ="1.0" encoding = "utf-8"?>
<Employee>
   <EmpDetails>
      <EmpId> 101 </EmpId>
      <EName> Raju </EName>
      <Designation> Programmer </Designation>
      <DOJ> 01/01/2014 </DOJ>
      <Salary> 15000 </Salary>
      <Deptno> 10 </Deptno>
   </EmpDetails>
      |
      |
      |
      |
      |
   <Dept>
      <Deptno> 10 </Deptno>
      <DName> coding </DName>
      <Location> Hyderabad </Location>
   </Dept>
      |
      |
      |
</Employee>
```

In VS.Net Xml parser and editor are integrated together. So when we type the code we find expandable, collapsable symbols.

coding support for XML in .Net:-

coding XML support in .Net Framework.

- Namespace: system.xml
  - classes: XmlReader, XmlWriter
- System.Data
  - DataSet

XmlReader class:-

This class is used to read the data from the xml file. This class contains various properties and methods which are helpful to read the data from the xml file.

properties in XmlReader class:-

1. EOF
2. HasValue
3. Name
4. NodeType
5. value
6. valueType

1. EOF :-

This property will return true, if end of file is encountered when reading the data. otherwise will return false.

② Hasvalue:-

This property will return true, if there is a value in child node otherwise will return false, if there is no value in child node.

③ Name:-

This property will return name of the tag. Tag might be opening or closing, parent or child or root.

④ NodeType:-

This property is used to identify type of the node, that is being read by xml Reader class object. All types of nodes avialable in an enumeration known as "xml node type" like.

XmlNodeType
- comment
- Document
- Element
- EndElement
- Text
- WhiteSpace, ...

⑤ Value:-

This property will return the attribute values or child tag values when reading the data.

⑥ valueType:-

This property will return, the type of the value that we are reading by using read method.

Methods with XML Reader class:-

(1) close()

(2) Read()

① close :-

This method is used, to close a opened xml-file into xml Reader class object.

② Read:-

This method is used to read the data from the xml file, Read method is reading the data in forward only and sequential only manner.

Read method will return true, if it encounters any tag or data. otherwise it will return false if it is encountering EOF.

Note:-

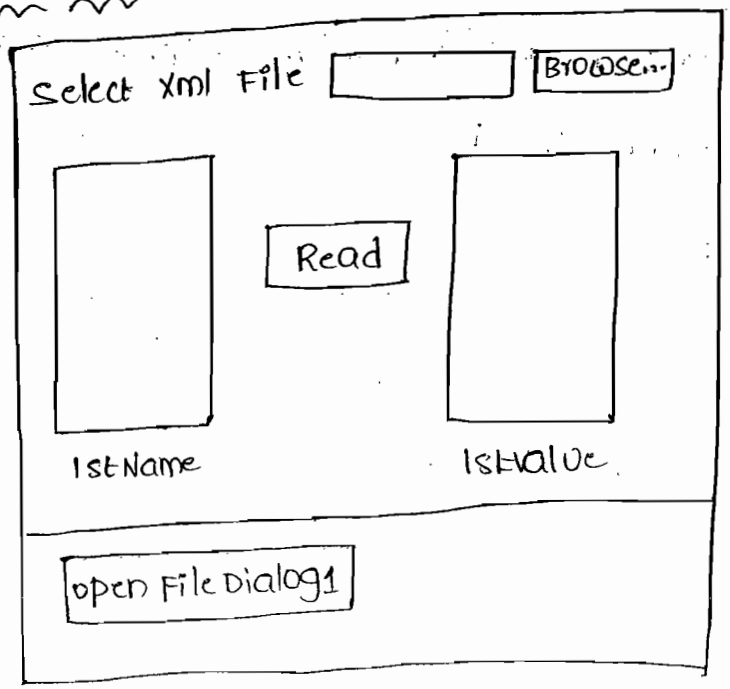
Those who are showing 2+ Years experience, before 4.0 .Net Framework will support other xml code classes. Those are

① xml Text Reader
② xml Text Writer

} must and should learn about this.

Note:- xml Query languages are ① xPath ② xQuery.

Example with xml Reader class :-



code:- Using system.Xml

```
private void btnBrowse_click(----)
{openFileDialog1.Filter = "XmlFiles|*.xml|| AllFiles|*.*";
 openFileDialog1.ShowDialog();
 txtName.Text = openFileDialog1.FileName;
}
private void btnRead_click(--)
{ XmlReader XR = XmlReader.create(txtName.Text);
  While(XR.Read()){
   {
    lstName.Items.Add(XR.Name);
    lstvalue.Items.Add(XR.Value);
   }
   XR.close();
}}
```

In the above code there is a drawaback avialable Every time Null values added to listbox. To overcome this we write the code like

```
private void btnRead_click(-----)
{
  XmlReader XR = XmlReader.create(txtName.Text);
  While(XR.Read())
   {
    if(XR.NodeType == XmlNodeType.Element)
       lstName.Items.Add(XR.Name);
    if(XR.NodeType == XmlNodeType.Text)
       lstvalue.Items.Add(XR.Value);
   }  XR.close();
}}}
```

Working with xml Writer class:-

This class is used to write the required data into the xml file.

→ This class contains properties and methods which are helpful to write the data into xml file.

Methods with writer class:-

1. writeEndDocument()
2. writeEndElement()
3. writecomment(String commentData)
4. writeStartElement(String ElementName)
5. writeStartDocument()
6. writevalue(AnyDataType Data).

1. writeEndDocument():-
   This method is used to write the end of file into the xml file

2. writeEndElement():-
   This method is used to write the closing tag into the xml file

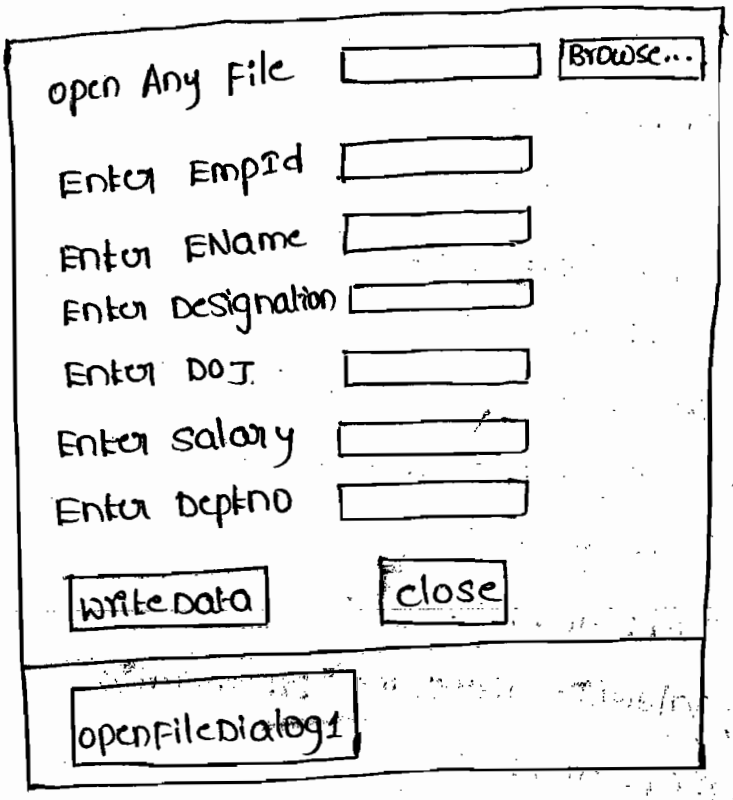
3. writecomment(String commentData):-
   This method is used to write the required comment value into the xml file.

4. writeStartElement(String ElementName):-
   This method is used to write the opening tags into the xml file.

5. writeStartDocument():-
   This method is used to write the begining statement or line into the xml file.

⑥ WriteValue (AnyDataType Data):-
This method is used to write the required data into the xml-file.

Example with xml writer class:-



code:-
```
using System.Xml
XmlWriter XW;
private void btnOpen_click(----)
{
openFileDialog1.Filter = "Xml Files|*.xml|AllFiles|*.*";
openFileDialog1.ShowDialog();
txtName.Text = openFileDialog1.FileName;
XW = XmlWriter.Create(txtName.Text, System.Text.Encoding.UTF8);
XW.WriteStartDocument();
XW.WriteStartElement("Employee");
MB.Show("Xml File open & Reading to write the data");
}
```

```
private void btnwriteData_click(---)
{
  XW.WriteStartElement("EmpDetails");
  XW.WriteStartElement("EmpId");
  XW.WriteValue(c.ToInt32(txtEmpId.Text));
  XW.WriteEndElement();
  XW.WriteStartElement("EName");
  XW.WriteValue("txtName.Text");
  XW.WriteEndElement();
  XW.WriteStartElement("Designation");
  XW.WriteValue("txtDesignation.Text");
  XW.WriteEndElement();
  XW.WriteStartElement("DOJ");
  XW.WriteValue(c.ToDateTime("txtDOJ.Text"));
  XW.WriteEndElement();
  XW.WriteStartElement("salary");
  XW.WriteValue(c.ToDouble("txtsalary.Text"));
  XW.WriteEndElement();
  XW.WriteStartElement("DeptNO");
  XW.WriteValue("c.ToInt32(txtDeptNO.Text));
  XW.WriteEndElement();
  XW.WriteEndElement();
  MessageBox.Show("Record is written into XML File");
}
private void btnclose_click(---)
{
  XW.WriteEndElement();
  XW.WriteEndElement();
  MB.Show("File is closed");
}
}
```

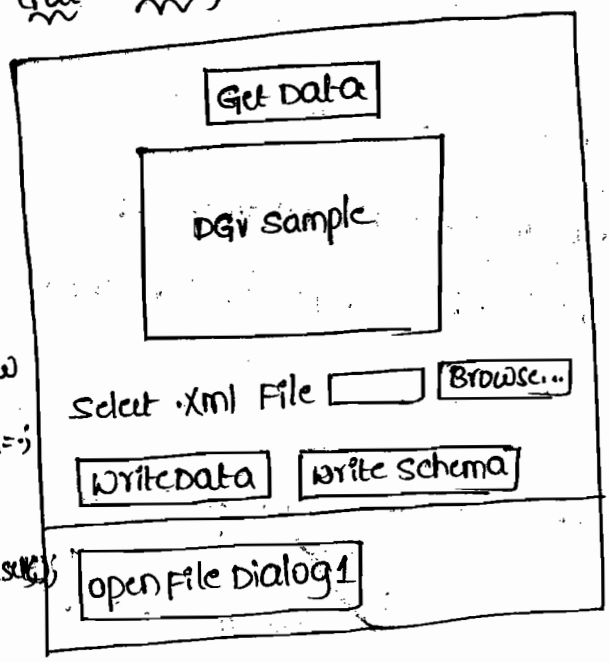
Working with Dataset:-

Dataset class contains following methods, to read the data into xml-file and to write the data into xml file.

1. ReadXml (String FileName)
2. ReadXmlSchema(String FileName)
3. WriteXml (String FileName)
4. WriteXmlSchema(String FileName);

Example to write the data using dataset into xml File:-

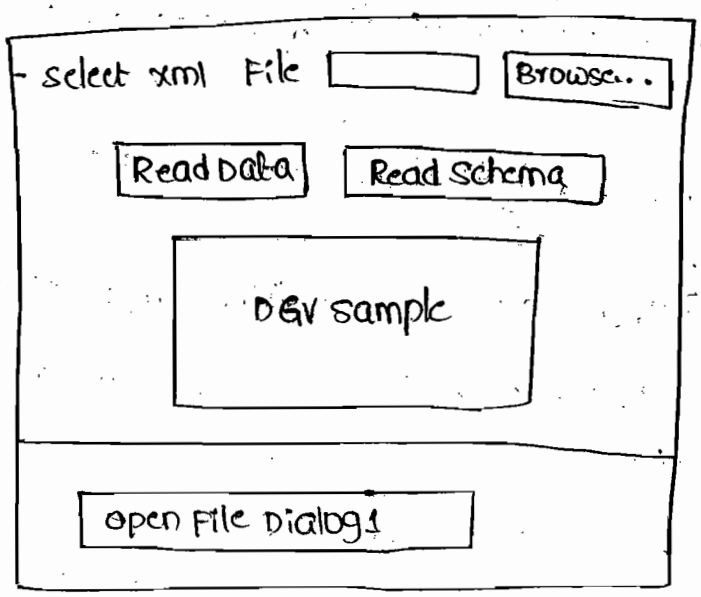
```
using System.Data;
using System.Data.
      SqlClient;
using System.Xml;
{ SqlConnection con = new
    SqlConnection("server=;
    ----)
  SqlDataAdapter Da;
  Dataset DS = new Dataset();
```




```
protected void btnGetData_click(----)
{
  String s = "select * ~~from EmpDetails~~"
  ~~SqlCommand cmd = new SqlCommand(s,con);~~
  Da = new SqlDataAdapter ("select * from EmpDetails, con);
  Da.Fill (DS, "EmpDetails");
  Da.FillSchema( DS, SchemaType.Source, "EmpDetails");
  DGVsample.DataSource = DS.Tables[0];
}
```

```
private void btnBrowse_Click(---)
{
  openFileDialog1.Filter = "xml Files | *.xml | All Files | *.*";
  openFileDialog1.ShowDialog();
  txtName.Text = openFileDialog1.FileName;

}
private void btnwriteData_click(--)
{
  DS.WriteXml(txtName.Text);
  MB.show("Dataset data has written into xml file");
}
private void btnwriteschema_click(--)
{
  DS.WriteXmlSchema(txtName.Text);
  MB.show("schema has written into xsd File");
}
}
```

Example with Read xml and Read xmlschema:-




```
private void btnOpen_click(---)
{ openFileDialog1.Filter = "xml Files | *.xml | All Files | *.*";
  openFileDialog1.ShowDialog1();
  txtName.Text = openFileDialog1.FileName;
}
```

```
private void btnRead Data_click (----)
{
    DS.Readxml(txtName.Text);
    DGVsample.Datasource = DS.Tables[0];
}
private void btnRead schema_click(-----)
{
    DS.ReadXmlSchema(txtName.Text);
    DGVsample.Datasource = DS.Tables[0];
}
}
}
```

→ Here write to code for to get the data and fill into dataset from xml file.

04/01/2015

CRYSTAL REPORTS

Reports:-

A report is an output tool which is used to display the required information to the user in required format.

→ A report always will contain summary information.

→ Report is always useful for the top level managements.

→ complete top level management will depend only on reports. to take appropriate decisions ~~to~~ related to their business needs.

→ In every real time Projects, reports are treated as separate module.

→ A report in general will contains 7 elements.

(1). Report Header
(2). Page Header
(3). Group Header
(4). Details Section
(5). Group Footer
(6). Page Footer
(7). Report Footer

(1). Report Header :-

This part will contain the information like company name, address, Quality levels, contact numbers, website address, etc---

(2). Page Header:-

This Part will contain the report ~~tey~~ tittle, field headings to display any information related to the Page like comments, etc.--

(3). Group Header:-

This part will contain the field information on which the data is grouped together.

(4). Details Section:-

This part will contain the original records information that is to be displayed within the report.

(5). Group Footer:-

This part will contain summary information for the group like sum of salaries for that dept, count of employees in the dept, average salaries in the dept, etc--

(6). Page Footer:-

This part will contain complete summary information for the page like sum of the salaries for the Page, current Page Number, total number of pages, etc---

(7). Report Footer:-

This part will contain complete summary information for the report like Grand total of salary prepared By, Authorized By, Date and Place, etc--

Sample Report without Grouping:-

Durga software solutions ①
Hyderabad

Employees Details Report ②

EmpId	EName	Designation	DOJ	Salary	Deptno
101	xxx	xxxx	xxx	xxx	xxx
102	xxx	xxxx	xxx	xxx	xxx
103	xxx	xxxx	xxx	xxx	xxx
⋮	⋮	⋮ ④	⋮	⋮	⋮

Sum of salaries for Page xxx
⑥ 1 of 10

⋮	⋮	⋮	⋮	⋮	⋮

Grand Total : xxx

Prepared By : xxx Authorized By: xxx
Date : xxx ⑦
Place : xxx

Here sample report without grouping contains only, the following

① Report Header
② Page Header
③ Detail Section
④ Page Footer
⑤ Report Footer.

Sample report with Grouping :-

① Durga Software solutions
Hyderabad

Employees Details Report ②

③ Deptno 10

EmpId	EName	Designation	DOJ	Salary
102	xxx	xxxx	xxx	xxx
④ 105	xxx	xxxx	xxx	xxx
106	xxx	xxxx	xxx	xxx

⑤ sum of salaries for Deptno 10 xxx

③ Deptno 10

101	xxx
104	xxx
108	xxx
⋮	

⑤ sum of salaries for Deptno 20 xxx

⑥ sum of salary for the Page : xxx

⑦ Grand Total : xxx

Prepared By : xxx Authorized By : xxx

Date : xxx

place : xxx

Reporting Tools:-

The softwares or tools used to develop the reports are known as reporting tools.

→ we find following reporting tools in IT field.

1. MS Data Reports
2. crystal Reports
3. SSRS
4. Hyperion
5. SSAS
6. BI Tools.

Crystal Reports:-

→ crystal reports originally developed with the name quick reports marketed by crystal Inc.

→ Originally this was developed for an accounting company for their accounting needs. Later this was purchased by c-gate, Next c-gate and crystal Inc was merged and rebranded as "crystal decisions". Now currently this is owned by SAP company

How it is related with microsoft Products:-

In 1990's crystal reports were used as a major reporting tool by integrating with visual basic. Though microsoft developed MS Data Reports it was n't as efficient as crystal reports. So companies were using only crystal reports as major reporting tool.

In 1998, 1999, 2000 microsoft purchased license of crystal reports from c-gate company for 10 years, and integrated in .Net completely. This was continued upto 2008 and framework 3.5 version. Mean while microsoft developed another reporting tool i.e., SSRS. from .Net framework 4.0 and Visual Studio 2010 microsoft is n't supporting crystal reports. Rather we should download separately crystal reports, download and use it.

To work with crystal reports, we need to use the following steps.

Step 1 : create the Datasource

Step 2 : Design the report

Step 3 : Display the report

crystal reports software download Link:-

http:// www.sap.com/solutions/sme/software/analytics/crystal-visual-studio/index.html

Example to create a report without grouping:-

→ create a new windows form application with the name WAReports.

Step 1 : creating the Datasource:-

→ create some xsd. file from the database ~~using~~ for. Empdetails table

Step 2 : Designing the report:-

→ Goto solution Explorer, select the solution, click with RH Button, click on Add, click on New Item. Select Reporting option of

visual c# Item from installed templates.

→ Select crystal reports template, type the crystal report name crystalreport1.rpt, click on add.

→ select the report type as standard, click on ok.

→ Double click on create new connection, Double click on ADO.Net (XML)

→ Click on Elipses Button, select xsd file, click on finish.

This will create a dataset with EmpDetails table.

→ Select EmpDetails table, add to selected tables option, click on Next.

→ select required fields from avialable fields add to fields to display option, click on Next, second time click on Next, Third time click on Next

→ select the report style as standard, click on finish.

→ This is created a report with the name crystalReport1.

Step3 : Display the report to the user:-

→ Goto windows form

→ Goto tool box, Goto reporting tool tab

→ Double click on crystal report viewer tool

→ Goto form1 code view, write the following code

```
using System.Data;
namespace WAReports
{
public partial class Form1 : Form
{
sqlconnection con = new sqlconnection("server=.; User Id=sa;
                Password= sa123; DataBase = Employee");

SqlDataAdapter Da;
DataSet DS;
CrystalReport1 objRPT = new CrystalReport1();
Public Form1 ()...
private void Form1_Load(---)
{ Da = new SqlDataAdapter("select * from EmpDetails",con);
DS = new DataSet();
Da.Fill (DS,"EmpDetails");
ObjRPT.setDataSource (DS.Tables[0]);
CrystalReportViewer1.ReportSource = objRPT;
}
}
}
```

→ Run the application and check you get the following error

could not load file or assembly 'file:///c:\Program Files\SAP BUSINESS-Objects\CrystalReports for .NET Framework 4.0\common\SAP BUSINESSobjects Enterprise XI 4.0\win32_x86\dotnet1\crdb_adoplus.dll' or one of its dependencies. The system cannot find the file specified.

To rectify this error, use the following steps

→ Goto open Windows Explorer

→ Goto following location.

c:\ProgramFiles\SAP Business objects\ crystal Reports for .NET Framework 4.0 \common\SAP Business objects Enterprise XI 4.0\ Win32_x86.

→ create a folder with the name dotnet1

→ copy the file crdb_adoplus.dll, paste into that folder.

→ Run the application you find the following error.

mixed mode assembly is built aganist version 'V2.0.50727' of the runtime and cannot be loaded in the 4.0 runtime without additional configuration information.

→ To rectify this error use the following steps.

→ goto solution Explorer, Double click on APP.config file in startup tag use following attribute

<startup useLegacyV2RuntimeActivationPolicy = "true">.

→ Run the application and check.

Example to create report with grouping:-

→ Goto solution explorer select the solution

→ click with RM Button

→ click on Add, click on New item

→ select reporting option from visual c# items, of installed templates.

→ select crystal report template

→ type the reportname crystal report2(crystalReport2.rpt)

→ select the crystal report type as standard, click on OK

→ Double click on my connections, from avialable data sources.

→ select Empdetails table, add to "selected tables option"

→ Click on Next, select all field from avialable fields add to field to display option, click on Next

→ select the fields deptno, add to group by option.

→ Click on Next, in summarized field you find all the fields that belongs to number type.

→ Delete Empid and Deptno fields from Here, keep only salary field click on Next, Second time click on Next, third time click on Next, Fourth time click on Next

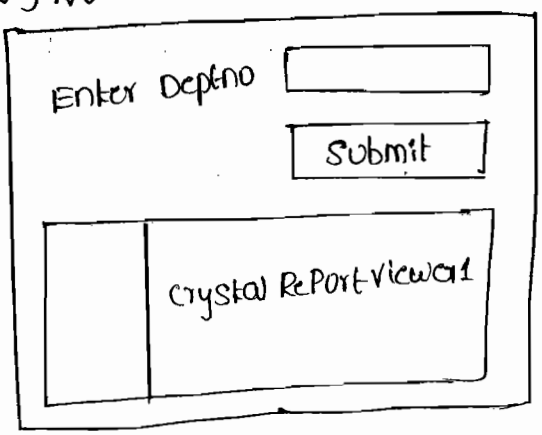
→ select the report style as standard, click on finish.

→ create a new windows form, Form2.

→ create crystal report code, write the following code

```
using system.Data;
namespace WABasics
{
 public partial class Form2: Form
 {
   SqlConnection con
   SqlDataAdapter Da;
   DataSet DS;
   crystalReport2 objRPt = new crystalReport2();
   public Form2()
   { Initializing component;
   }
   private void Form2 - Load (---)
   {
     Da = new SqlDataAdapter ("select * from EmpDetails", con);
     DS = new DataSet();
     Da.Fill(DS, "EmpDetails");
     objRPt.SetDataSource (DS.Tables[0]);
     crystalReportViewer1. ReportSource = objRPt;
   }
 }
}
```

Example to pass parameters to crystal reports :-



→ create a new windows Form

→ Design the form

→ create a new crystal report without grouping or crystal report1, Goto form code and write the following code

```
Using System.SqlClient;
namespace WAReports
{
 Public Partial class Form1 : Form
 {
  sqlconnection con = new sqlconnection("server=';------);
  sqlDataAdapter Da; Dataset DS;
  crystalReport1 objRPt = new crystalReport1();
  public Form3()...
  private void Form3_Load(---)
  {
   GetData("select * from EmpDetails");
  }
  private void GetData(String s)
  {
   Da = new sqlDataAdapter(s,con);
   DS = new Dataset();
   Da.Fill (DS,"EmpDetails");
   objRPt.SetDataSource (DS.Tables[0]);
   crystalReportViewer1.ReportSource = objRPt;
  }
  private void btnSubmit_click(----)
  {
   String s = " select * from EmpDetails where Deptno =
                                    "+txtdeptno.Text;
   GetData(s);
  }
```

creating charts in crystal Reports:-

→ Goto crystal report1

→ goto report header, click with RH button

→ click on insert, click on chart, click on type tab page

→ select type of chart as bar,

→ click on Data tab page, select for each record in place of on change of, select the field EName, Add to for each record option.

→ select the field salary, add to show values option.

→ Select text tab page, type chart tittle & x-axis tittle, Y-axis tittle.

→ click on ok, run the application and check

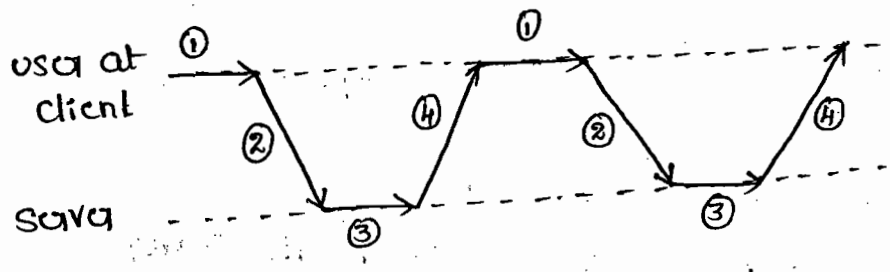
02/01/2015

AJAX

[ASYNCHRONOUS JAVASCRIPT AND XML]

To understand AJAX we look into synchronous and asynchronous programming models.

Synchronous Programming Model:-



Step1 : user at client prepares a request

Step 2 : Request is sent to server

Step3 : server will process the request

Step4 : Response is delivered to the client.

Drawbacks in synchronous Programming model:-

(1). Burden on the webserver is more

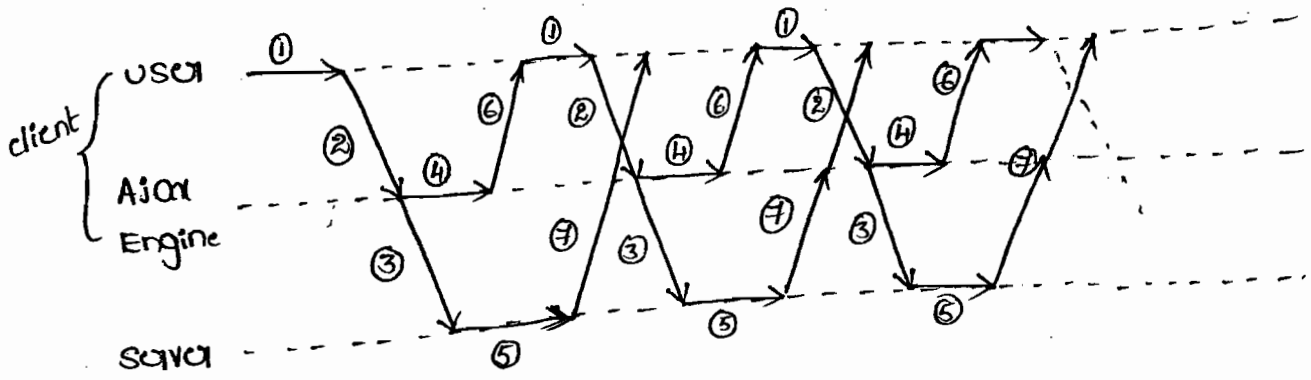
(2). user ideal time is more

(3). Payload over the network will be more

(4). overall application Execution is slow.

To overcome such drawbacks, asynchronous Programming model is used.

Asynchronous Programming Model:-



Step1 : User at client side will Prepare a request.

Step2 : Request is sent to Ajax engine

Step3 : Ajax engine will send the request to the server

Step4 : Ajax engine will Process some kind of request at client side.

Step5 : Server will Process client request.

Step6 : Ajax engine at client side will give response to the client.

Step7 : Server will send the response to the client. But before server response, here client will Prepare next request to the server.

Advantages in Asynchronous Programming Model:-

1). Burden on the webserver is less.

2). Payload coming over the network is less.

3). User ideal time is less.

4). overall application Performance is improved.

→ An Ajax is combination of technologies, n't a single technology.

→ AJAX means Asynchronous JavaScript And XML

→ The term AJAX is coined (introduced) by J.J. Garret.

→ In the year 2005, the AJax technology or framework is also introduced by, once JJ. Garret introduced this many companies and people introduced so many frameworks in support to Ajax technology. To support client side and server side scripting like

JavaScript

1. BackBase
2. DOJOToolkit
3. Ext
4. MOOTools
5. JQuery
6. Prototype
7. Script.aculo.us
8. yahoo UI Library

Java

1. BackBase
2. Echo
3. Google WebTool Kit
4. DWR

.Net

1. AJax.Net Professional
2. ASP.Net Ajax

PHP

1. SAJAX
2. XAJAX
3. TigerMouse.

AJax.Net Professional:-

→ originally developed by codeplex.com

→ microsoft completely moved to open source from AJax.Net - professional

→ That means it is developed by codeplex community.

ASP. Net Ajax:-

ASP. Net Ajax is developed by microsoft with code name as "ATLAS" in 2005.

→ ASP. Net Ajax Architecture is, spread over client side and server side

→ ASP. Net Ajax will provide rich and ~~ready~~ very good UI element

components of ASP. Net Ajax:-

1. HTML + HTML DOM
2. CSS
3. Java script
4. JSON
5. XML HttpRequest

ASP. Net ajax will contain 5 components are above shown.

1. HTML + HTML DOM:-

→ HTML is used for designing purpose at client side.

→ HTML DOM is nothing but document object model. The data is represented in the form of hierarchy

→ All HTML elements are accessed in the form of objects by using which it is possible to set or get the properties of elements at runtime.

② CSS :-

we know that CSS are used to apply attractive styles to the webpages and elements of the webpages.

→ CSS are introduced in the year 1997 in HTML 4.0 version.

③ JavaScript:-

we know that JavaScript is used to implement validations and events at client side

④ JSON (JavaScript Object Notation):

JSON is introduced by Douglas Gracford in the year 1995.

→ JSON is most helpful in implementing "Light-weight Data Exchanging formats."

→ JSON can be used independently or can be integrated with any of the server side or client side technologies.

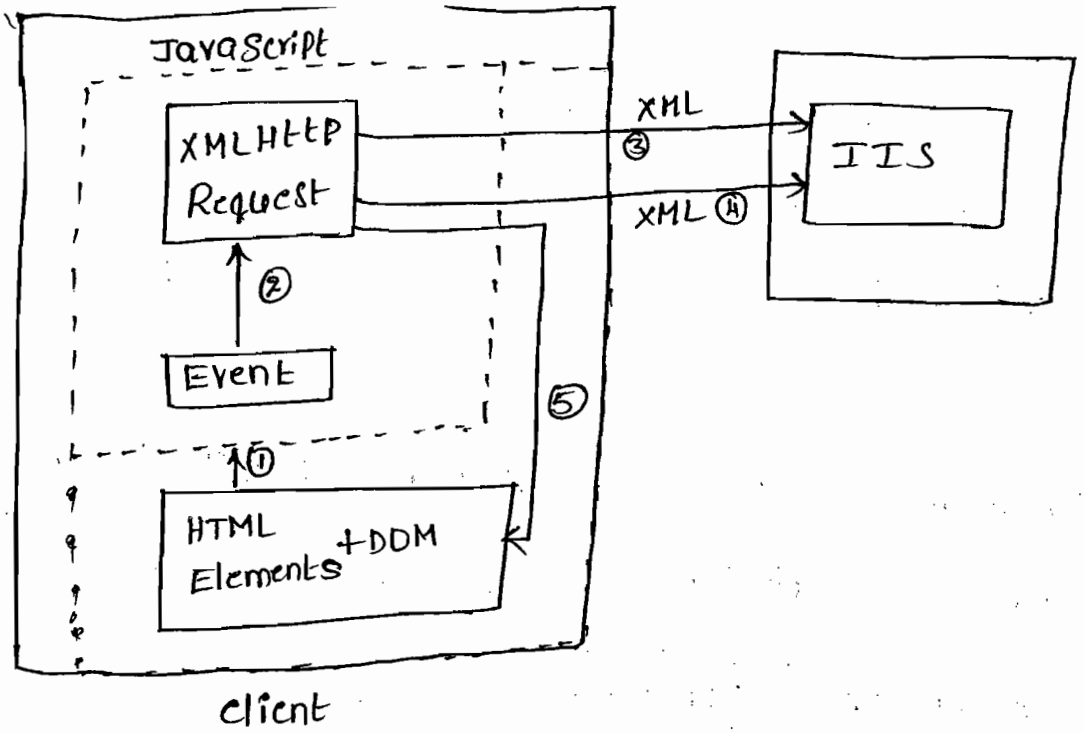
→ JSON file will have a default Extension of ".json".

→ By using JSON we can also Provide Namespace mechanism, object based mechanism to the JavaScript.

→ JSON will Provide all object oriented Programming facilities to the JavaScript

⑤ XMLHttpRequest object:-

This is the crucial component in Ajax, which is helpful in converting the request and responce into XML form.



Step1: some Html element at client side will raise a request or event in the form of Html

Step2: This event is passed to XHR object. i.e., XML HTTP Request Object

Step3: XHR object will convert this request into xml form and sends to ~~client~~ server.

Step4: Request is processed at server side, response is delivered to client in the form of xml

Step5: XHR object at client side will convert this back to Html and JavaScript form and will be given to client

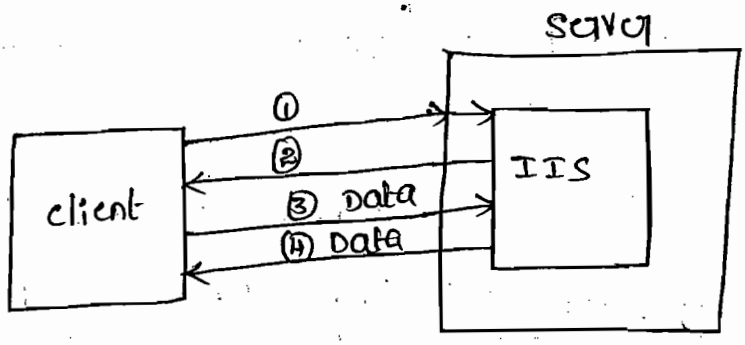
05/01/2015 Programming models in Ajax:-

Ajax will support 2 types of programming models,

(1). Server Centric Programming Model

(2). client centric Programming model.

(1). Server Centric Programming Model:-



Step1 : client sends for the request for first time for a page

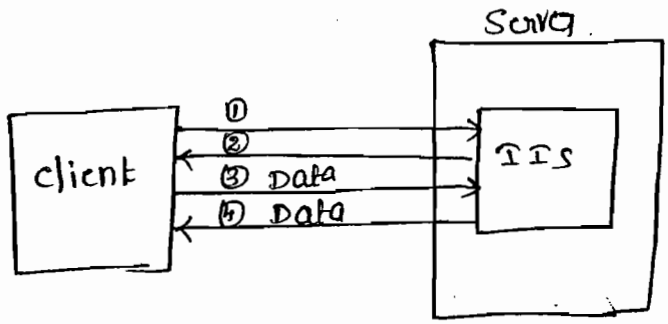
Step2 : client request will be executed at server side initial rendering and UI behaviour will be sent to client. But logic code of the application will remain at server side only, will n't be sent to client

Step3 : PostBack Request will goto server, logic code will run at server side

Step4 : The result along with updated UI rendered information will be sent to client.

In server centric Prog. model, complete logic code will remain at server and executes at server side only

②, client centric Programming Model:-



Step1:- Client sends request first time for a Page, to the server

Step2: Server will Process the request and will send response to the client, along with logic code of the Page

Step3: complete logic code will run at client side only if at all required. request will be sent just for the data to the server

Step4: Server will respond along with data to the client, Here always from Postback request logic code will ~~remain at~~ execute from client only

Advantage in client centric Programming model:-

1) Execution will be faster when compared with server centric Programming model

Working With Ajax controls:-

Asp.Net Ajax supports following controls

1. ScriptManager
2. ScriptManagerProxy
3. UpdatePanel
4. UpdateProgress
5. Timer

1. ScriptManager:-

This control is used to Provide, ajax features for the webPage. When this control is included in any webPage then, the Page will Provide all ajax features, otherwise ajax features will n't Provided for the Page

2. ScriptManagerProxy:-

This control will Provide the facilities to access the web Services or WCF Services from the client side directly with ajax features

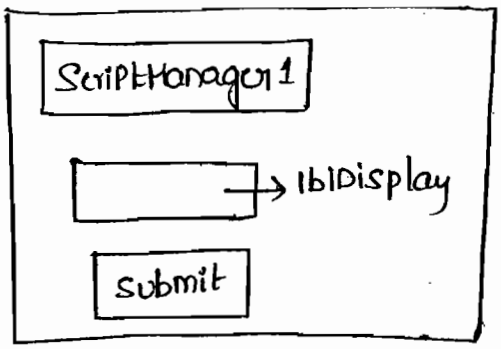
3. UpdatePanel:-

This control is used, to Provide client centric Programming model. i.e, any controls that are Placed inside the Update Panel will work with Client centric Programming model, controls that are Placed, outside the Update Panel will work with Server centric Programming model

Example with Server centric Programming Model:-

→ create a new website with the nam AJax Example

→ Design the Page, write the following code



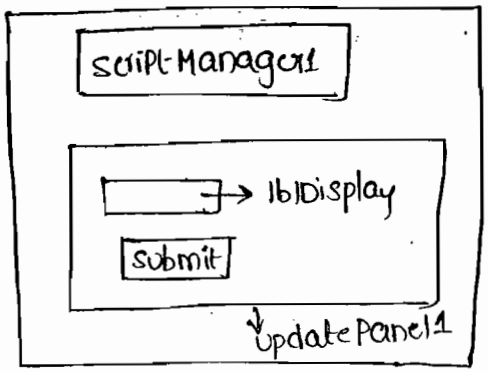

```
protected void Page_Load(---)
{
}
protected void btnsubmit_click(---)
{ lblDisplay.Text =" welcome";
}
```

→ Run the application, at first request will be sent to server but logic code will remain at server, when user clicks on submit button, request will goto server and code is Executed at server and response is sent to client.

Making same Example with client centric Programming model:-

→ create a new webpage

Design the webpage and write the following code



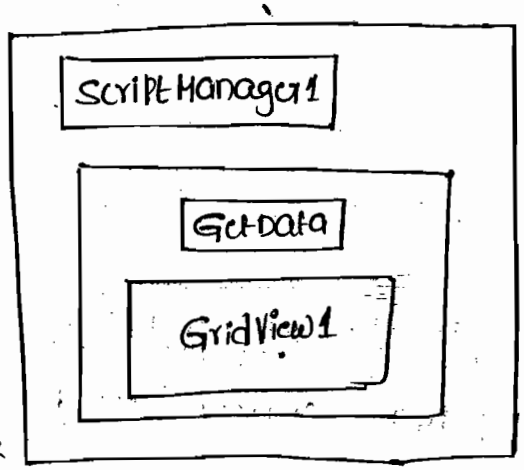

```
protected void btnsubmit_click(---)
{ lblDisplay.Text ="welcome";
}
```

→ Run the application, at first request, request will be executed at server side, response is delivered along with logic code to the client.

When user clicks on submit button, request will n't goto server, rather will be executed at client side only.

Example with client centric Programming model for database Data:-

→ create a new webpage, Design the webpage, write the following code for GetData Button




```
protected void btnGetData_click
(---)
{
    SqlConnection con = new SqlConnection(-----)
    SqlDataAdapter Da = new SqlDataAdapter ("select * from EmpDetails",
                                            con);
    Dataset DS = new DataSet();
    Da.Fill (DS, "EmpDetails" );
    GridView1.DataSource = DS.Tables[0];
    GridView1.DataBind();
}
```

→ Run the application and check when user clicks on get data button, request will n't be submitted to server, rather will be executed in client itself.

⑤ Timer Control:-

This control is used, to execute the code repeatedly again and again without any User Intervention

properties with Timer Control:-

① Enabled → True (Default), False

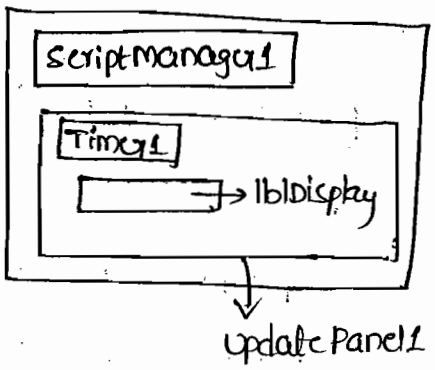
② Interval → 8000 (Default)

Default Event of timer control:-

Default event of timer control is "Tick"

Example with Timer control:-

(Displaying current time to the User)



→ create a new webpage, Design the webpage, set the following properties to the timer.

① Enabled → True

② Interval → 1000

Write the following code for Tick event

```
protected void Timer1_Tick(---)
{
    lblDisplay.Text = DateTime.Now.ToString();
}
```

→ This is most suitable with client centric Prog. model

→ If this is made with server centric, every time after every second, request will sent to server, executes at server side

and response delivered to client, which is n't suitable and will reduce the application performance.

(4) UpdateProgress control:-

This control is used to display, any kind of message to the user when request is being processed.

properties with updateProgress control:-

① AssociateUpdatePanelId

② DisplayAfter → 500 (Default)

① Associate ~~with~~ UpdatePanelId:-

used to set or get, Id of the update panel control, that is to be attached with update Progress.

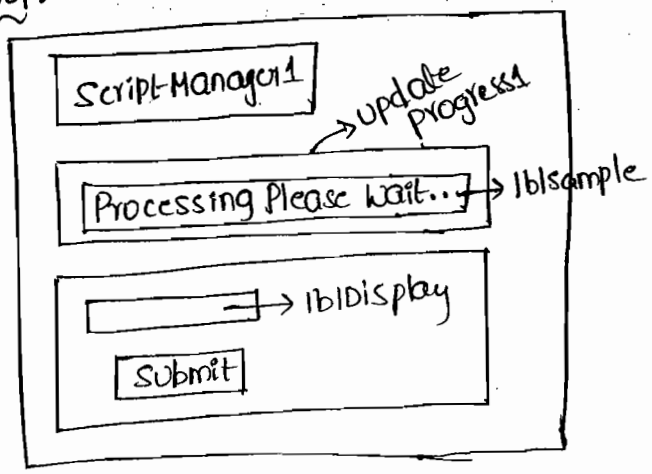
② DisplayAfter:-

used to set or get the time period after which the update panel control should be displayed

Example with UpdateProgress Control:-

→ create a new webpage, Design the webpage, In update Progress create a label, type the text processing please wait.

→ In submit button write the following code



```
protected void btnSubmit_Click(---)
{
    System.Threading.Thread.Sleep(10000);
    lblDisplay.Text = "welcome";
}
```

→ Run the application and check

06/01/2015

→ Download toolkit link is www.asp.net/ajax

Ajax Control Toolkit:-

Ajax control tool kit, is developed by code-plex.com, which is an open source community from microsoft. This can be downloaded from following link www.asp.net/ajax.

When we download this, we download zip file, extract this zip file into a folder with the name tool kit, then open Vs.Net

→ goto toolbox, click with RM Button

→ click on Add, click on tab, type Ajax Tool Kit

→ click with RM Button, click on choose items. click on Browse

→ goto the location where ajax control toolkit is available.

→ Select the "dll", click on open, click on ok.

→ this will add all controls to the toolbox

→ Toolkit contains 2 types of controls.

(1). Extender controls

(2). Non-Extender controls

→ All ajax controls (Extender & non-Extender) working with client centric Prog.mod

(1) Extender controls:-

These controls can't work independently, rather they will be attached to other controls. They help the other controls to provide different kind of facilities.

(2). Non-Extender Controls:-

These controls will work independently, they can help other controls or can be used independently.

List of Extender Controls in Tool Kit:-

S.NO	control Name
01	Always visible Control Extender
02	Animation Extender
03	Autocomplete Extender
04	BalloonPopupExtender
05	calendar Extender
06	Cascading DropDown
07	CollapsiblePanel Extender
08	ColorPicker Extender
09	Confirm Button Extender
10	Drag Panel Extender
11	DropDown Extender
12	Dropshadow Extender
13	DynamicPopulate Extender
14	Filtered TextBox Extender
15	HoverMenu Extender

16. ListSearch Extender
17. Masked Edit Extender
18. Modal PopUp Extender
19. MultiHandleSlider Extender
20. Mutually Exclusive CheckBox Extender
21. Numeric Updown Extender
22. Paging Bulleted List Extender
23. Password Strength
24. PopUpControl Extender
25. Resizable Control Extender
26. Rounded Corners Extender
27. Slide Extender
28. Slideshow Extender
29. TextBox Water Mark Extender
30. Toggle Button Extender
31. UpdatePanel Animation Extender
32. Validator Callout Extender
33. Masked Edit Validator

List of Non-Extender Controls:-

(1) Accordion
(2). Accordion Content Panel
(3). Accordion Pane
(4). Ajax File Upload
(5). Asny File Upload
(6). Barchart
(7). Bubblechart
(8). comboBox
(9). Editor
(10). Gravatar
(11). JQueryScriptControl
(12). Linechart
(13). NOBOT
(14). Piechart
(15). Rating
(16). Recorder List
(17) ScriptControl Base

(18), Seadragon

(19), Tabcontainer

(20), ToolkitScriptManager

(21), Twitter

Working with Ajax control Toolkit controls:-

(1), confirmButtonExtender Control :-

This control is used to provide confirmation of the operation from the user

properties with confirmButtonExtender Control:-

(1), TargetControlId

(2), confirmText

(1), TargetControlId :-

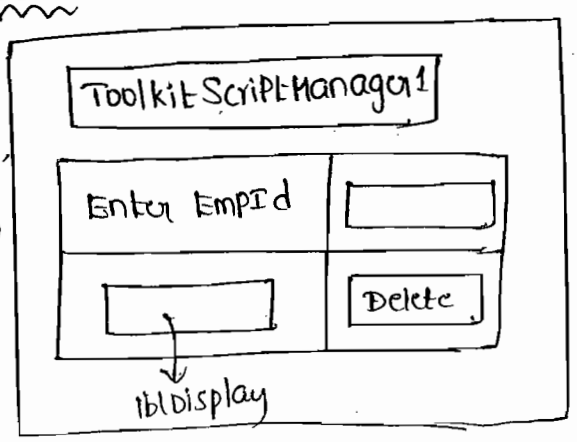
Used to set or get the required control id to which, the confirm button should be ~~Extra~~ extended control attached.

(2), confirmText:-

Used to set or get the required text to be displayed for the confirmation

Example with confirmButtonExtender:-

→ Here at first in Design, we can't use Ajax enabled controls, if you use it raises compilation error.



→ create a new webpage, Design the webpage

→ Goto tool box, toolkit tool tab, Double click on tool kit Script manager. This will create toolkitScriptmanager1

→ Goto source, double click on confirm button Extender control. This will create confirmButtonExtender1 and write the following code

```
<asp: confirmButtonExtender ID =" confirmButtonExtender1"
 runat= "server" TargetControlID= "btnDelete" confirmText= " Do you
 want to Delete" >
</asp: confirmButtonExtender>
```

→ Write the following code for delete button.

```
using System.Data;
Using System.Data.SqlClient;
SqlConnection con = new SqlConnection(----)
protected void btnDelete_click(---)
{
 String s =" select * from EmpDetails where EmpId "= + txtEmpId.Text.
 SqlCommand cmd= new SqlCommand(s,con);
 con.open();
 int i= cmd.ExecuteNonQuery();
 con.close();
 lblDisplay.Text = i+ " Row (s) Deleted";
 }
}
}
```

(29) TextBoxWaterMarkExtender Control:-

This control is used, to display the required text within the textbox with water mark feature

→ This control is used in general in searching facilities or to give any hints to the users.

properties with TextBoxWaterMarkExtender control:-

① TargetControlId

② WaterMarkCssClass

③ WaterMarkText

② WaterMarkCssClass:-

used to set or get, required css class, that is to be attached to the textbox control.

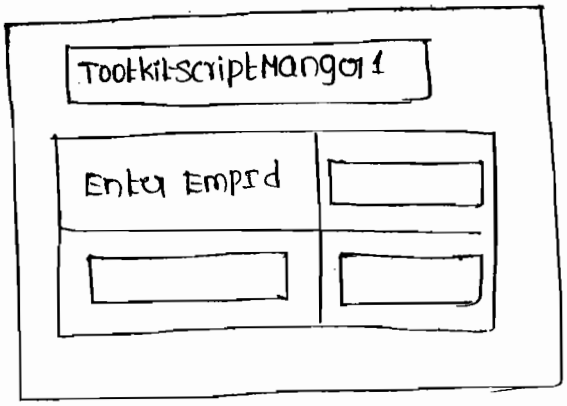
③ WaterMarkText:-

used to set or get, the required text, that is to be displayed within the textBox.

Example with TextBoxWaterMarkExtender Control:-

→ create a new webpage, design the webPage

→ Select TextBox, txtEmpId, click on smart tag button, click on add extender.



Select TextBoxWaterMarkExtender, Click on OK

→ Goto properties window, goto txtEmpId _ TextBoxWaterMarkExtender property.

In watermark Text, the data "Type Here"

In WaterMark CSS class, type "S1"

→ Goto source, write the following code for style

```
<style type = "text/css">
 .S1
 {
   color: gray;
 }
```

→ Run the application and check, and write code for deleting.

01/01/2015 (11) Filtered TextBoxExtender Control :-

This control is used to provide filteration facilities for the data that is to be accepted in to the textbox from the user.

properties with FilteredTextBoxExtender control :-

① TargetControlId

② FilterMode → validchars (default) / Invalidchars.

③ FilterType → custom / Numbers / UpperCase Letters / LowerCase Letters

④ validchars

⑤ Invalidchars.

① ~~Target control Id~~ (Filter mode) :-

used to set or get, weather the user entered data is valid or invalid.

FilterType:-

used to set or get value that indicates what type of data is to be accepted into the textbox.

(h). validchars:-

used to set or get list of the characters that are to be allowed into the textbox

Invalidchars:-

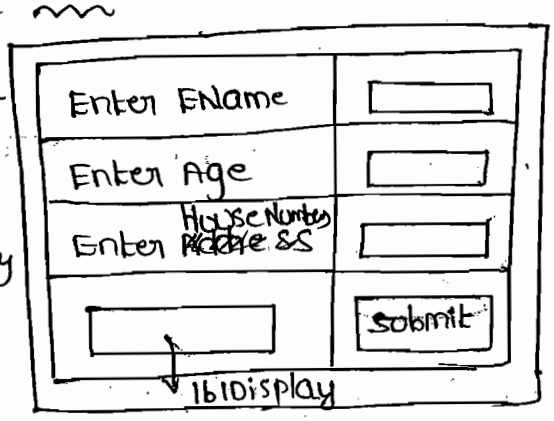
used to set or get list of the characters that are n't allowed into the textbox

→ when we set filter type to "custom", then valid and invalid chars come into Picture

Example with Filtered TextBoxExtenderControl:-

List of validations to be Implemented:-

1. txtName should accept letters only.
2. txtAge should accept numbers only
3. txtHouseNumber should accept Numbers, capital letters, - and /



→ create new webPage, Design the webPage

→ select textbox txtEName, click on smart tag button select filtered textbox extenderControl, click on ok, similarly attach for txtAge and txtHouseNumber filtered textbox extender controls, then goto source part and write the following code.

```
<asp: FilteredTextBoxExtender ID="txtEName_FilteredTextBoxExtender"
runat="Server" Enabled="True" TargetControlID="txtEName"
Filtermode="validchars" FilterType="LowerCaseLetters, UpperCaseLetters" />
</asp: FilteredTextBoxExtender>
:
<asp: FilteredTextBoxExtender ID="txtAge_FilteredTextBoxExtender"
runat="Server" Enabled="True" TargetControlID="txtAge"
Filtermode="validchars" FilterType="Numbers" />
</asp: FilteredTextBoxExtender>
:
<asp: FilteredTextBoxExtender ID="HNum_FilteredTextBoxExtender"
runat="Server" Enabled="True" TargetControlID="txtHNum"
Filtermode="validchars" FilterType="Numbers, UpperCaseLetters, Custom"
</asp: FilteredTextBoxExtender>
```

10) BaloonPopupExtenderControl :-

This control is used to display a baloon POPUP when user places mouse pointer or when focus enters into the control, etc.

properties with BaloonPopupExtender Control:-

① TargetControlID

② BaloonPopupControlID.

③ Baloonsize → small (Default)
→ medium
→ Large

④ BaloonStyle → Rectangle (Default), Cloud, Custom

⑤ CustomclassName

⑥ customCSSUrl

⑦ DisplayOnclick → True (Default), False

⑧ DisplayOnFocus → True, False (Default)

⑨ DisplayonmouseOver → True, False (Default)

⑩ Position → Auto, TopRight, TopLeft, BottomRight, BottomLeft

② BaloonPopupcontrolId:-

used to set or get the control Id for which message will be avialable to display in baloonPopup.

③ BaloonSize:-

used to set or get required size for the baloon, how the baloon should be displayed.

④ BaloonStyle:-

used to set or get ~~for the~~ required style. for the baloon.

⑤ customclassName:-

used to set or get the required CSS class name to be applied when baloonSize is set to custom.

⑥. CustomCSSUrl:-

Used to set or get, the required Url address to apply as CSS when baloon size is set to custom.

⑦. Display onclick:-

When set to true, when user clicks on, the control the baloonPopup Extender will be displayed.

When set to false, when user clicks on the control the baloonPopup Extender will n't be displayed.

⑧. Display onclick:-

↳ when set to true, when control gets the focus, baloonPopup Extender will be displayed.

↳ when set to false, when control gets the focus, baloonPopup Extender will n't be displayed.

⑨. Display on mouseOver:-

↳ when set to true, baloonPopup control will be displayed when mouse is placed on the control.

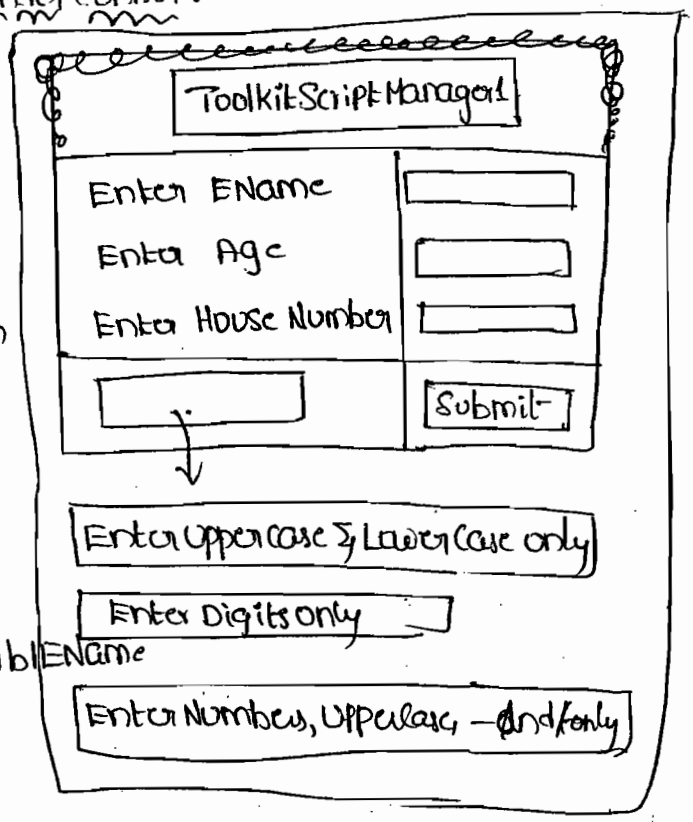
↳ when set to false, baloonPop control will n't be displayed when mouse is placed on the control.

⑩. Position:-

Used to set or get the required position where popup control should be displayed.

Example with BaloonPOPUPExtender Control :-

→ select txtEName, click on smart tag Button, Click on add Extender, click on baloonPop Extender, Click on ok.



→ Goto properties window Set the following properties

1. Baloon POPUP ControlId → lblEName
2. BaloonSize → Small
3. Baloon style → Cloud.
4. Display onclick → False
5. Display on mouse → True

select the textBox txt age, Click on smart tag, click on add Extender. Select baloonPopupExtender, Goto properties window and set the following properties.

1. BalbonPOPUP controlId → lblAge
2. Baloon ~~style~~ Size → small
3. Baloon style → Rectangle
4. Display onclick → False
5. Display on mouse → True

select the textBox txtHNum, Click on Smart tag, Click on add extender, Select baloonPOPUP Extender and set the following properties

1. Baloon control Id → lblHNO
2. Baloon Size → small
3. Baloon Style → cloud
4. Display on click → False
5. Display on mouse → True

(05). CalenderExtenderControl :-
This control is used to display the calender to the user

① TargetControlId
② Animated → True (Default), False
③ ClearTime → True, False (Default)
④ cssClass
⑤ DaysModeTitleFormat → MMMM, YYYY
⑥ DefaultView → Days (Default), Months, Years
⑦ Format
⑧ PopupButtonId
⑨ PopupButtonPosition → BottomLeft (Default), BottomRight, TopLeft, TopRight, Right, Left
⑩ SelectedDate
⑪ StartDate
⑫ TodaysDateFormat
⑬ EndDate

① Animated :-
↳ when set to true, calender control is displayed with animation facility
↳ false, calender control is n't displayed with animated facility

③ ClearTime :-
↳ when set to true, Time format will be displayed in calender
↳ false, Time format will n't be displayed in calender

④ CssClass :-
↳ used to set or get required css class id to apply the style.

5. DaysmodeTitleFormat:-

used to set the required format, to be displayed within the calender control

6. DefaultView:-

when set to Days, calender control is displayed within the days-format

when set to months, calender control is displayed within the months-format

~~Format:-~~

years: calender control is displayed with years format.

7. Format :-

used to set or get the required format for the date within the calender control.

8. PopupButtonId:-

used to set or get the required control Id when user clicks on the calender should be displayed.

9. PopupButtonPosition:-

used to set or get required Position, where calender control is to be displayed.

10. SelectedDate:-

used to set or get current date within the calender control.

11. StartDate:-

used to set or get a value that indicated from where date value should begin in calender control.

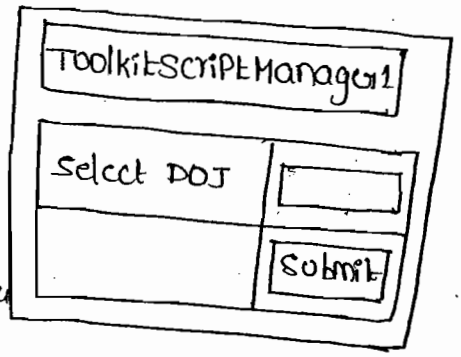
(12) **TodaysDateFormat:-**

used to set or get the required format for the todays date.

(13). **EndDate:-**

used to set or get the closing date upto where the calender control should be displayed

08/1/15 **Example with calendarExtender control:-**

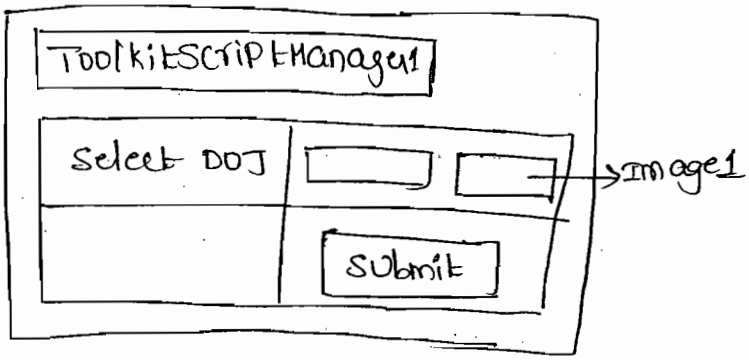


→ create a new webpage, Design the webpage, select the textBox txtDOJ

→ Click on smart tag button, click on AddExtender, select the calendarExtender control, click on OK.

→ Run the application and check.

If you would like to display calender when user clicks on image button use the following mechanism



→ create a new webpage

→ Design the webpage

→ select textbox txtDOJ

→ click on smart tag button, click on Add Extender, ~~click on~~ Select calender Extender control, click on OK

→ Goto properties window, goto PopupButtonId, Type Image1

→ Now run the application and check, when user clicks on image button, calender control will be displayed now.

(23). PasswordStrengthControl:-

This control is used to specify the strength of the Password when user is entering the Password into the Password textbox.

properties with PasswordStrengthControl:-

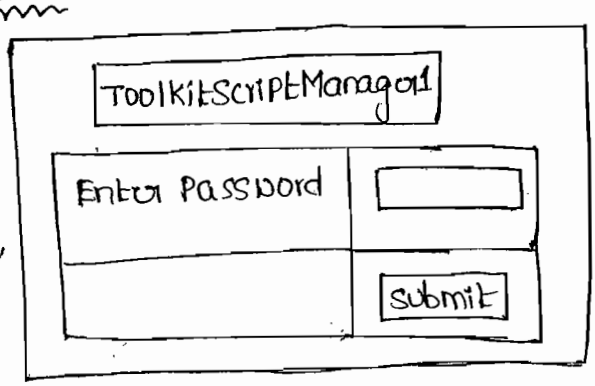
1. TargetControlId
2. BarBoarderCssclass
3. BarIndicatorCssclass
4. Calculationweightings
5. HelpStatusLabelId
6. minimumLowerCasecharacters
7. minimumUpperCasecharacters
8. minimumNumericCharacters.
9. minimumSymbolcharacters
10. PreferredPasswordStrength
11. PrefixText
12. StrengthIndicatorType → Text (Default), Bar
13. StrengthStyles.
14. TextCssclass
15. TextStrengthDescriptions
16. TextStrengthDescriptionStyles

Example with PasswordStrength Control:-

→ create a new webPage

→ Design the webPage

→ Goto source, Goto toolbox, goto toolkit tool tab



→ Double click on password strength control, write the following code.

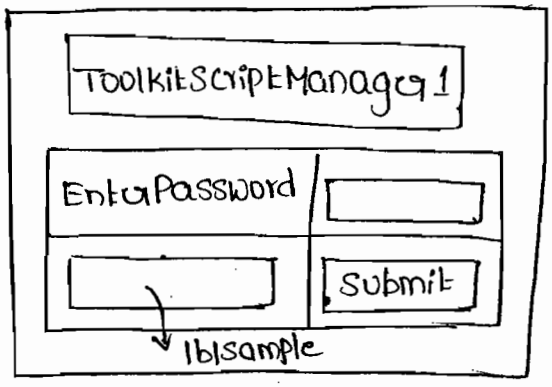
```
<asp: PasswordStrength ID = "PasswordStrength1" runat = "server"
TargetControlID = "txtPassword" minimumLowerCasecharacters = "3"
minimumUpperCasecharacters="3" minimumSymbolcharacters ="3"
```

minimumNumericCharacters="3" PreferedPasswordLength="12"

PrefixText="your password is:" TextStrengthDescriptions="verypoor; poor; Average; OK; Better; Best; Strong; Unbreakable">

</asp:PasswordStrength>

Example:2



→ create a new webPage

→ Design the webPage

→ Goto source, goto toolbox

→ Goto toolkit tool tab.

→ Double click on PasswordStrength control

→ write the following code

<asp:PasswordStrength ID="PasswordStrength1" runat="server"

TargetControlId="txtPassword" StrengthIndicatorType="Bar indicator"

BarBorderCssClass="S1" BarIndicatorCssClass="S2"

minimumLowercasecharacters="3", minimumUppercasecharacters="3"

minimumNumeric characters="3" minimumSymbolcharacters="3"

PreferredPassWordStrength="12" HelpStatusLabelID="lblSample"

</asp:PasswordStrength>

CSS code:-

```
<style type="text/CSS">
 .S1
 {
    boarder-Style : solid;
    boarder-color : Blue;
    height : 20PX
    width : 300PX
```

• S2
{
color : green
background - color : green
}

creating Ajax website for checking all controls Help:-

→ Goto Ajax toolkit folder, you find a zip folder, copy all that zip folder files and Paste into c: Ajax website folder. [created folder]

→ Goto IIS, create a virtual directory, map to ajax website folder

→ Goto VS.Net click on file, click on open, click on website

→ Select Ajax website from Local IIS, click on open,

→ Run the application and check.

WebHostings :-

(1). purchase Domain
(2). purchase webspace
(3). Host/Deplay the application

→ Best website to purchase domain is www.Godady.com.

→ murthy - 8374435962
(HCL → Mahesh Babu sir friend)

09/1/15

N-TIER ARCHITECTURE

In general any code we write any application we develop related to database will contain 3 types of code.

(1). UI Design code.

(2). Business Logic Code

(3). Data Access code.

UI Design & validations code

This code is related to creating user interface elements like textbores, Buttons, checkBoxes, radiobuttons, etc..

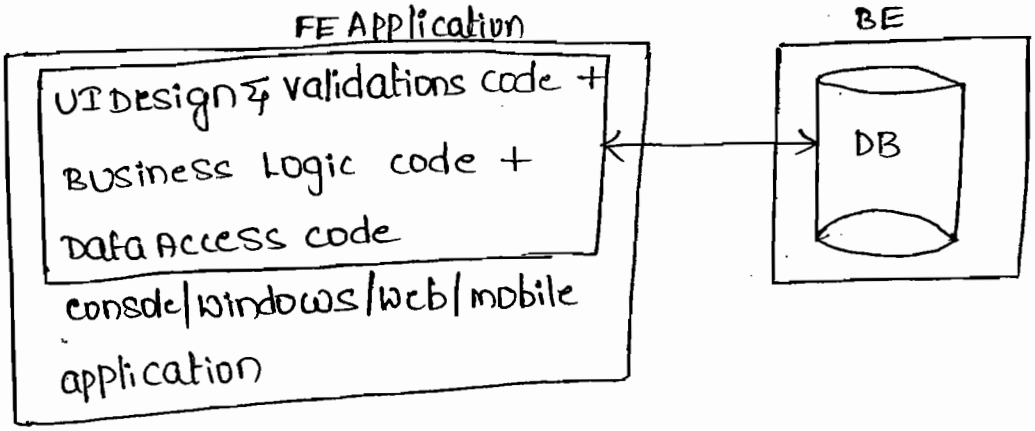
In general, this code will change based on the type of application that we develop. i.e., For console application one type of code, for Windows applications some type of code, For web application some type of code, etc..

(2). Business Logic Code:-

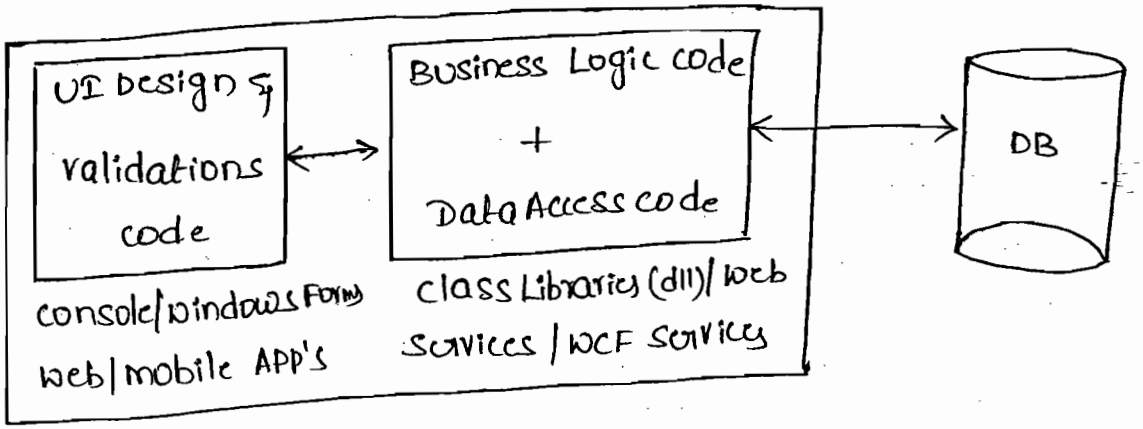
This code is related to the business nature of the domain that we computarized. Based on the domain that we are computarizing For hospital domain one type of code for banking domain one type of code. etc...

(3). Data Access Code:-

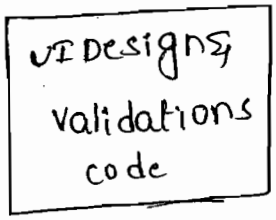
This code is related to the interacting with DB, like getting the data to DB, updating the data to DB etc...


FE Application
UI Design & validations code +
Business Logic code +
Data Access code
console/windows/web/mobile
application
BE
DB

2-Tier


UI Design &
validations
code
console/windows Form
web/mobile APP's
Business Logic code
+
Data Access code
Class Libraries (dll)/ web
Services / WCF Services
DB

3-Tier


UI Design &
validations
code