

SATHYA TECHNOLOGIES

B.kannaBabu

C#.NET NOTES

SRI RAGHAVENDRA XEROX

Software Languages Material Available

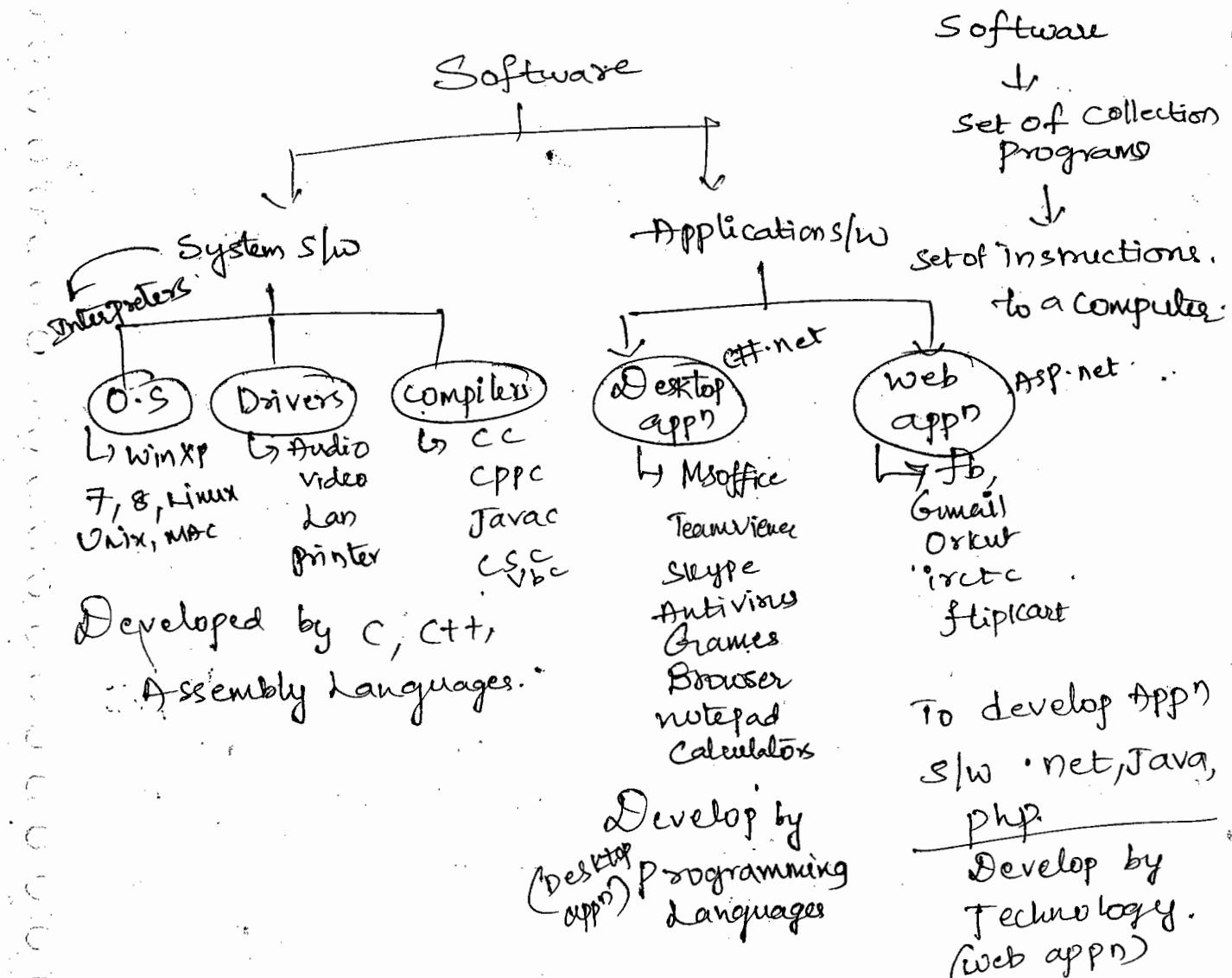
Beside Bangalore Ayyangar Bakery, Opp. C DAC, Ameerpet, Hyderabad.

Cell: 9951596199

C# .Net

.Net

1st June 2015



Programming Languages: C, C++, VC++, VB6.0, Java, C# .net, VB .net

Technologies: ASP .net, PHP, JSP, servlets.

Database: SQL Server, Oracle, MySQL.

World of .net

ASP - Active Server Pages
JSP - Java " "

Subscribe in YouTube
www.facebook.com/worldofdotnet

Angular JS

2nd June 2015

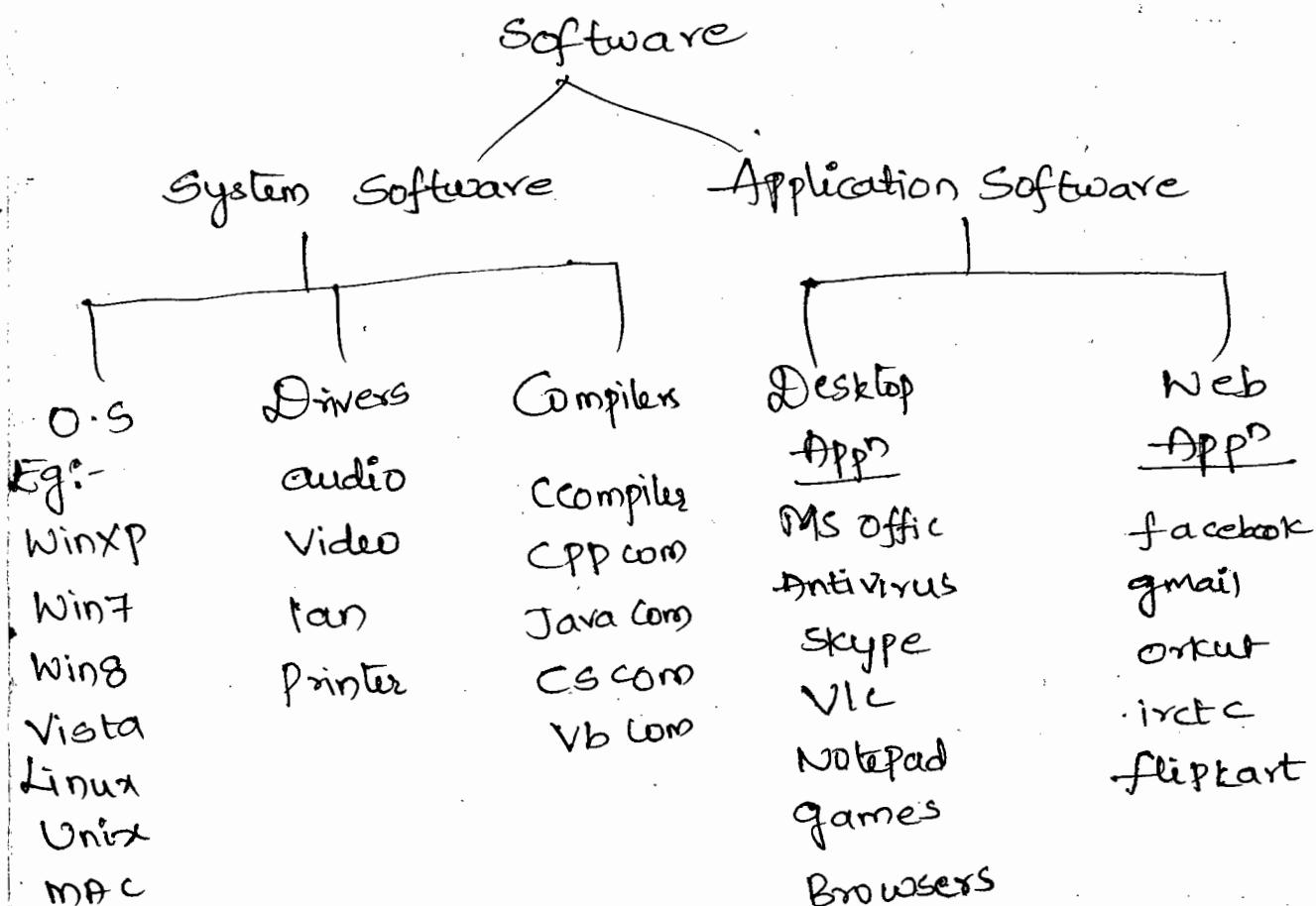
What is Software?

Software is collection of programs.

What is program?

Program is set of instructions given to a computer.

Software is basically divided into two types



System softwares can be developed by using C, C++ and Assembly languages.

What is Desktop app's?

The app's that was installed on users' desktop are called as Desktop app's.

Eg:- MsOffice, Skype etc.

→ Desktop app's can be developed by using programming languages like C#-net, VB-.net, Java

What is C#-net?

C#-net is a programming language which is used to develop desktop app's.

What is Web app's?

The app's that can be accessed by using Browser and Internet are called as Web app's.

Eg:- facebook, gmail etc.

→ We can develop web app's by using different technologies like Asp.net, PHP, JSP etc.....

What is Asp.net?

Asp.net is a technology which is used to develop web applications.

The full form of Asp.net is Active Server Page. Network enable technology.

C#-net → C Sharp .net

(or)

C #(ash).net

Desktop Application

1. DA must be installed on user's desktop.
2. DA can be developed by using programming languages like C# .net, VB.net, Java.
3. Internet is not mandatory for DA.
4. We cannot access Desktop app by using Browser.
5. DA are single-user applications.
6. Web Server is not required.
7. DA cannot host on server.
Eg:- MS office, Skype etc.

Web Applications

- WA cannot install web applications.
- WA can be developed by using technologies like Asp.net, JSP, PHP.
- Internet is mandatory for WA.
- Browser is mandatory to access WA.
- WA are multi-user application.
- Web Server is mandatory.
- WA must host on server.
- Eg:- Facebook, Gmail etc.

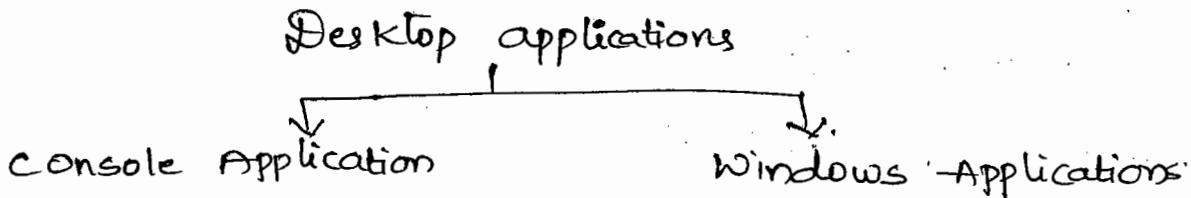
www.facebook.com/groups/worldofdotnet

| 3rd June 2015

What is C#.net?

C#-net is a programming language which is used to develop desktop apps.

→ Desktop app's are of two types.



Console App's :- Console App's are also called as character user interface applications (CUI).

→ The user can interact with console app's through keyboard.

→ Console Apps are not user friendly.

Eg:- Socket programming.

→ The output of console app" will be on command

Windows Forms Applications :- Windows Forms Applications are also called as (GUI) Applications , Graphical User Interface Applications.

→ The user can interact with windows forms Apps by using keyboard & mouse.

→ These are user friendly.

Eg:- Notepad, MsOffice, Antivirus, skype, calculator etc.

Note:- We can develop Console appn' in C#-net

by using console appn template.

→ We can develop Windows forms app's in C# .net by using windows forms app template.

4th June 2015

ASP.NET :-

Asp.net is a technology which is used to develop web applications.

What is Web application?

The application that can be accessed with Browser and internet are called as Web applications.

What is a Website?

Website is a collection of web pages.

Eg:- www.gmail.com.

www.axisbank.com

www.irctc.com.

What is a webpage?

Webpage is collection of information like images, icons, advertisements etc.

Eg:- LoginPage, inbox, SentItems, drafts.

Enter Username

Enter Password

Sign in

Login Page

In order design the above login Page we have to learn html,

HTML :- HTML is Hypertext markup language which is used to design the webpage.

The output of any web application will be on browser.

What is web browser ?

Web browser is a software which is installed on user's desktop. Client → Browser in Web apps

Eg:- Internet explorer, Mozilla firefox, Google chrome etc.

What is Java script ?

Java script is a scripting language which is used to perform Client side Validations and animations.

Eg:- Username must not be empty } Validations.
Password must not be empty } . . .

Username and password must not be same.

Password and confirm password must be same.

Phonenumbers must accept only numbers.

Password must not exceed 6 characters.

Age must be between 18 and 25.

What is CSS ?

CSS is Cascading Style sheets.

CSS is used to apply the style for the

Controls. (or) webpages like font, height, width color, background colour, border colour etc.

ASP.NET = HTML + Javascript + CSS + C# + .NET

What is a web server?

An individual running machine is called as server.

What is Web Server?

Web server is a software which is installed on server machine

Eg:- Tom cat , Apache, IIS (Internet Information Service)

→ IIS is a webserver for all the Asp.net web apps.

Web servers → technology

```
graph LR; IIS[IIS] --> ASPNet[ASP.NET]; ApacheTomcat[Apache, Tomcat] --> JSP[Java JSP, Servlets]; Apache[Apache] --> PHP[PHP]
```

PHP - & also called as WAMP (or) LAMP

WAMP → Windows O.S., Apache, MySQL, PHP.
∴ (OS) (Webserver) (Database) (Tech)

LAMP → Linux, Apache, MySQL, PHP.

Go Daddy is a used to buy domain space.
on web server.

(5th June 2015)

Microsoft Visual Studio editor :-

It is a development environment which is used to develop different types of Applications in .net Supportable languages and technologies i.e if we want to develop any desktop applications or web applications we have to use visual studio editor.

* Desktop applications can be developed by using programming languages like C#-net or VB-.net.

* Web applications can be developed by using Asp-.net.

* By using Visual Studio editor, we can develop different applications like

1. Console applications
2. Windows Forms applications
3. Web applications
4. Class library
5. WPF Application
6. WCF Application
7. MVC Applications.

What is .net?

.Net is a framework which is used to run the programs that was developed by using Microsoft visual studio editor.

⇒ .Net frame is a runtime environment which is used to run the applications that was developed by using different languages and technologies like C#-net, VB-.net, Asp-.net etc.,

Visual studio Versions

MSVS - 2000

MSVS - 2003

MSVS - 2005

MSVS - 2008

MSVS - 2010

MSVS - 2012

MSVS - ~~2013~~

- 2000 -

- 2003 -

- 2005 -

- 2008 -

- 2010 -

- 2012 -

- 2013 -

.Net F/w Versions

.net 1.0

.net 1.1

.net 2.0

.net 3.5

.net 4.0

.net 4.5

.net 5.0

For every Version of Visual Studio editor Microsoft has released .net framework version.

- ⇒ The Applications that was developed in Visual Studio 2000 can run by using .net framework 1.0.
- ⇒ Whenever we install the latest version of the .net framework then automatically the older versions of .net frameworks will be installed.
- ⇒ It is not required to install .net framework Separately. When we install Visual studio editor then automatically .net framework will also be installed.

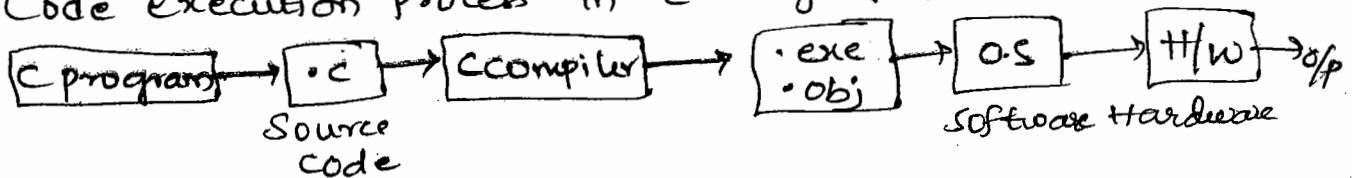
NO NOTES [6th June 2015]

6th June 2015

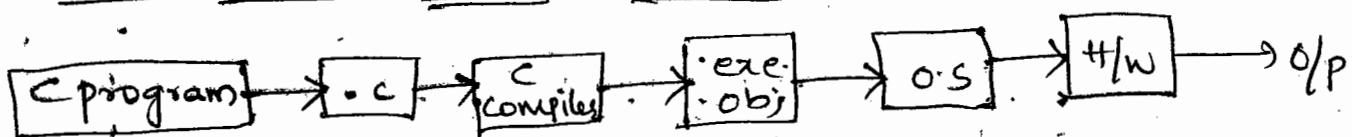
Four steps for a Program

Write → Save → Compile → execute.

Code execution process in 'C' language:-



Code Execution process in 'C' language:-



→ Every 'C' program must save with ·c extension

which is called as Source Code.

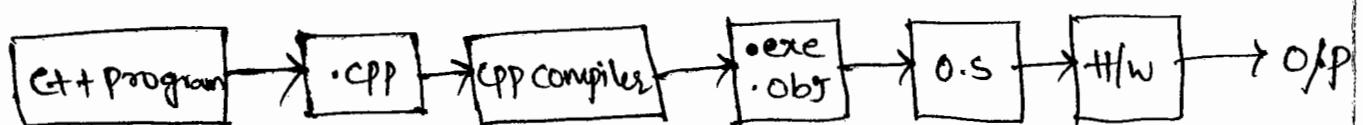
→ Whenever we compile 'c' program with c compiler

then the Compiler will check for Syntax errors.

→ If there are no syntax errors then the compiler will generate ·exe which internally consists of ·obj

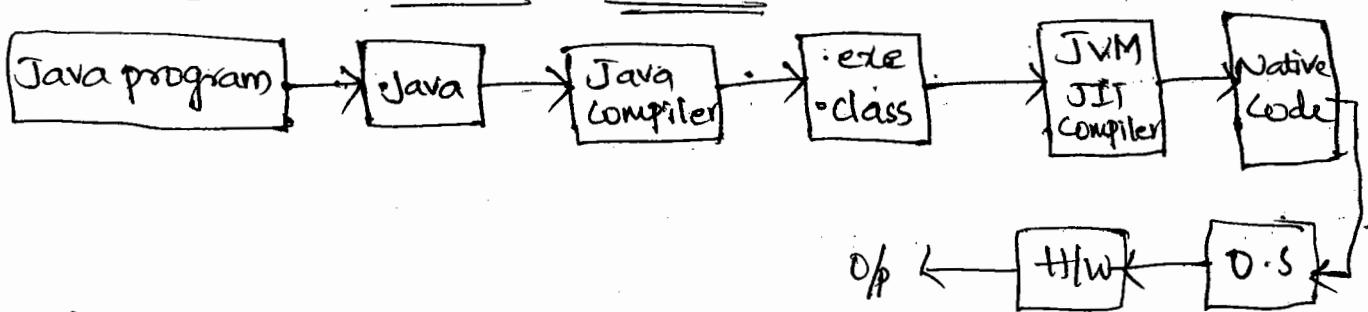
→ Finally when we execute the program ·obj can understand by O.S and O.S will communicate with Hardware to get the Output.

Code Execution process in 'C++' language:-



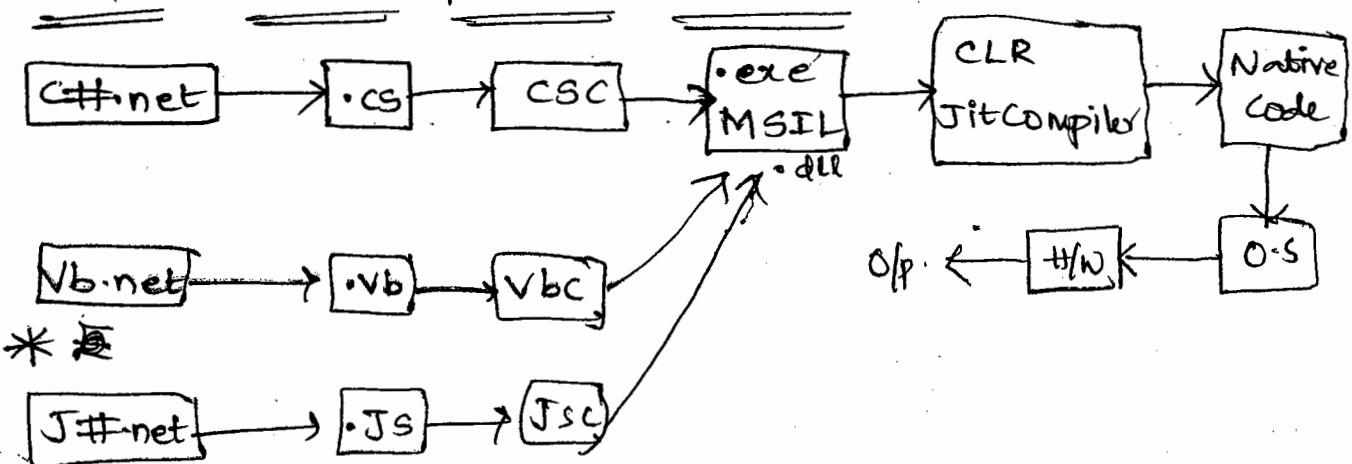
- * Every C++ program must save with .CPP which is called as source code.
- * Whenever we compile the program with CPP Compiler, the Compiler will check for Syntax errors.
- * If there are no Syntax errors it will generate .exe file.
- * When we execute the program O.S will communicate with CPU and Hardware and get the output.

Code execution process in Java :-



- * Every Java program must save with .Java extension.
- * Whenever we Compile the Java program, Java compiler will check for Syntax errors.
- * If there are no Syntax errors it will generate .exe file which internally consists of ".class" file.
- * When we execute the program within the JVM (Java Virtual Machine) JIT Compiler (Just-In-time) will Convert .class file into native code.
- * Finally O.S will communicate with Hardware and get the Output.

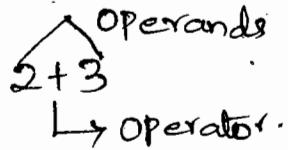
Code Execution process in .net :-



- * Every C#-net program must save with .cs extension which is called as source code.
- * Whenever we compile the program with CSC (C sharp compiler) then the Compiler will check for Syntax errors.
- * If there are no Syntax errors then the Compiler will generate .exe file which internally consists of MSIL code (Microsoft Intermediate Language).
- * Whenever we execute the program with CLR (common language runtime) CLR is heart and soul of .net Framework within the CLR, Jit compiler will convert MSIL code to native code and finally OS will communicate with HW to get output.
- * Any .net language will have their own Compilers and will have only one CLR.

9th June 2015

Operator:- An operator is used to perform operation on two or more operands.



What is an expression?

An expression is the combination of two or more operators.

$$\text{e.g. } 2+3-5*6$$

Whenever we solve an expression we have to solve based on the priority of the operator.

* Different types of operators are

* Arithmetic Operators +, -, *, /, %

* Relational " <, >, <=, >=, ==, !=

* Logical " &&, ||

* Assignment " =

* Increment/Decrement " ++, --

Arithmetic Operators:- These operators are used to perform arithmetic operations like addition, sub, multiplication and division.

+, -, *, /, %

/ - gives Quotient

% - gives Remainder

$$\frac{3}{2} = 1 \quad (\text{Quotient}) \quad 3 \% 2 = 2 \quad (\text{Remainder})$$

int + int = int

int + float = float

int * float = int

int * float =

int / int = int

int / float = float

What is the difference between / and %?

/ gives us Quotient and modulus will give remainder.

$$5/3 = 1$$

$$5 \% 3 = 2$$

Note:- If the numerator is less than denominator then Quotient is zero (0) and remainder is numerator.

* Whenever we solve an expression we have to solve the expression based on the priority of the operator.

1st priority $\Rightarrow \ast, /, \%$

2nd " $\Rightarrow +, -$

3rd " $\Rightarrow =$

$$2 + 3 * 5 - 3 * \frac{7}{6} - 2 * 3 + 2$$

$$2 + 15 - \frac{21}{6} - 6 + 2$$

$$\cancel{17 - \frac{21}{6} - 6 + 2} \quad \cancel{\frac{13}{6} + \frac{21}{57}}$$

$$\cancel{\frac{13}{7} - \frac{21}{6}} \\ = \frac{57}{6} = 9$$

$$6) \overline{57} (9$$

$$\rightarrow 2 + 3 * 5 - 3 * \frac{7}{6} - 2 * 3 + 2 \quad \frac{21}{6} = 3$$

$$2 + 15 - 3 * \frac{7}{6} - 2 * 3 + 2$$

$$2 + 15 - 3 - 6 + 2$$

$$2 + 15 - \frac{21}{6} - 2 * 3 + 2$$

$$= 10$$

$$2 + 15 - 3 - 2 * 3 + 2$$

$$= \underline{\underline{10}}$$

$$2 - 3 * \frac{7}{\frac{0}{2}} - 7 * \frac{2}{\frac{7}{0}} + \frac{\frac{2}{0}}{\frac{7}{0}} * 6 + 2 - 3$$

$$2 - \frac{21}{\frac{0}{2}} - 7 * \frac{2}{\frac{7}{0}} + \frac{2}{\frac{0}{7}} * 6 + 2 - 3$$

$$\Rightarrow \frac{21}{20} (10)$$

$$2 - 1 - \overbrace{7 * \frac{2}{\frac{7}{0}}} + \frac{2}{\frac{0}{7}} * 6 + 2 - 3$$

$$2 - 1 - \frac{14}{7} + \frac{2}{\frac{0}{7}} * 6 + 2 - 3$$

$$\Rightarrow 2 (1)$$

$$2 - 1 - 2 + \frac{2}{\frac{0}{7}} * 6 + 2 - 3$$

$$2 - 1 - 2 + 2 * 6 + 2 - 3$$

$$2 - 1 - 2 + 12 + 2 - 3$$

$$\cancel{30} + 12 + 2 - 3$$

$$-1 + 12 + 2 - 3$$

$$32 (1)$$

$$= \underline{\underline{10}}$$

$$\frac{2}{3} - \frac{3}{\frac{7}{0}} + \frac{3}{\frac{0}{9}} - \frac{9}{\frac{0}{11}} + \frac{11}{\frac{0}{121}} - \frac{121}{236} + \frac{236}{597} - 1$$

$$0 - \frac{3}{\frac{7}{0}} + \frac{3}{\frac{0}{9}} - - -$$

$$0 - 0 + \frac{3}{\frac{0}{9}} - -$$

$$6) \frac{7}{6} (1)$$

$$3 - 9 + 11 - 0 + 0 - 1$$

$$-6 + 11 - 1 \quad \overline{12 * 3}$$

$$= 4$$

$$6 - 2 + 0 * \frac{7}{\frac{0}{6}} - 32 + 7$$

$$6 - 2 + 0 * 32 + 7$$

$$\cancel{6} 11 - 32$$

$$= -21$$

$$Q = S \times D$$

Relational Operators (or) Comparison Operators :-

These Operators are used to compare two or more expressions or values.

- * Relational operators will always return a boolean value either True or false.
- * Relational operators/ comparison operators are used to check the condition. and that condition will always return a boolean value either true or false.

$$5 > 3 (\text{T}), 5 \leq 3 (\text{F}), 5 != 5 (\text{F}), 5 == 5 (\text{F})$$

$$5 \leq 2+3-5 (\text{F}), 5 != 5-5 (\text{T})$$

$$2-3 * \frac{5}{7} + \frac{7}{3} * 7 - \frac{9}{81} + \frac{81}{121} >$$

7) 15 (2)

$$2 - \frac{15}{7} + \frac{7}{3}$$

$$-\frac{13}{7} + \frac{49}{3} - \frac{7}{21} + \frac{21}{183}$$

$$2-2 + \frac{7}{3} * 7 - \frac{9}{81} + \frac{81}{121}$$

$$-1 + 16 - 0 + 0 \\ 15$$

$$2-2+2 * 7$$

$$2-2+14 - \frac{9}{81} + \frac{81}{121}$$

$$2-2+14 - 0 + 0$$

$$= \cancel{14} \quad 14 \quad \frac{81}{\cancel{81}} \quad \frac{9}{81}$$

$$14 \geq 15 (\text{F})$$

$$7 + \frac{49}{7} + 7 * \frac{7}{7} \rightarrow 7 + 7 * 7$$

$\Rightarrow 49 + 7$

$$7 - 7 + \frac{49}{7} - 7 + 7 * 7$$

$$\frac{0}{7}$$

$$0 + 0 - 7 + 49$$

"it"

$= 42$

$5 > 5$ (T)

~~$-10 \geq 5$ (F)~~

Note:- When we solve an expression it will return a value.

* When we solve a condition it will always return a boolean value.

10th June 2015

Conditions.

		Conditions.	
		c ₁	c ₂
		&&	
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

Logical Operators:- These operators are to check two or more conditions and return a boolean value true or false.

$$2+3-5 \geq 10*6 - \frac{2}{3} + \frac{7}{9} \text{ and } 9-3*\frac{5}{2} - 7*11 \leq -21+20$$

$$0 \geq 60 \text{ and } 9-7-7*11 \leq -4$$
$$\begin{array}{r} 9-7 \\ -77 \\ \hline -75 \end{array}$$

$\therefore (\text{F}) \text{ and } (\text{T}) \Rightarrow (\text{F})$

60

$9 - \frac{15}{2}$

2) $\frac{15}{2}$

$$2 \geq 6, \text{ and } 2 \leq 5, \text{ and } 2 \geq -2$$

(f) and (T) and (T)

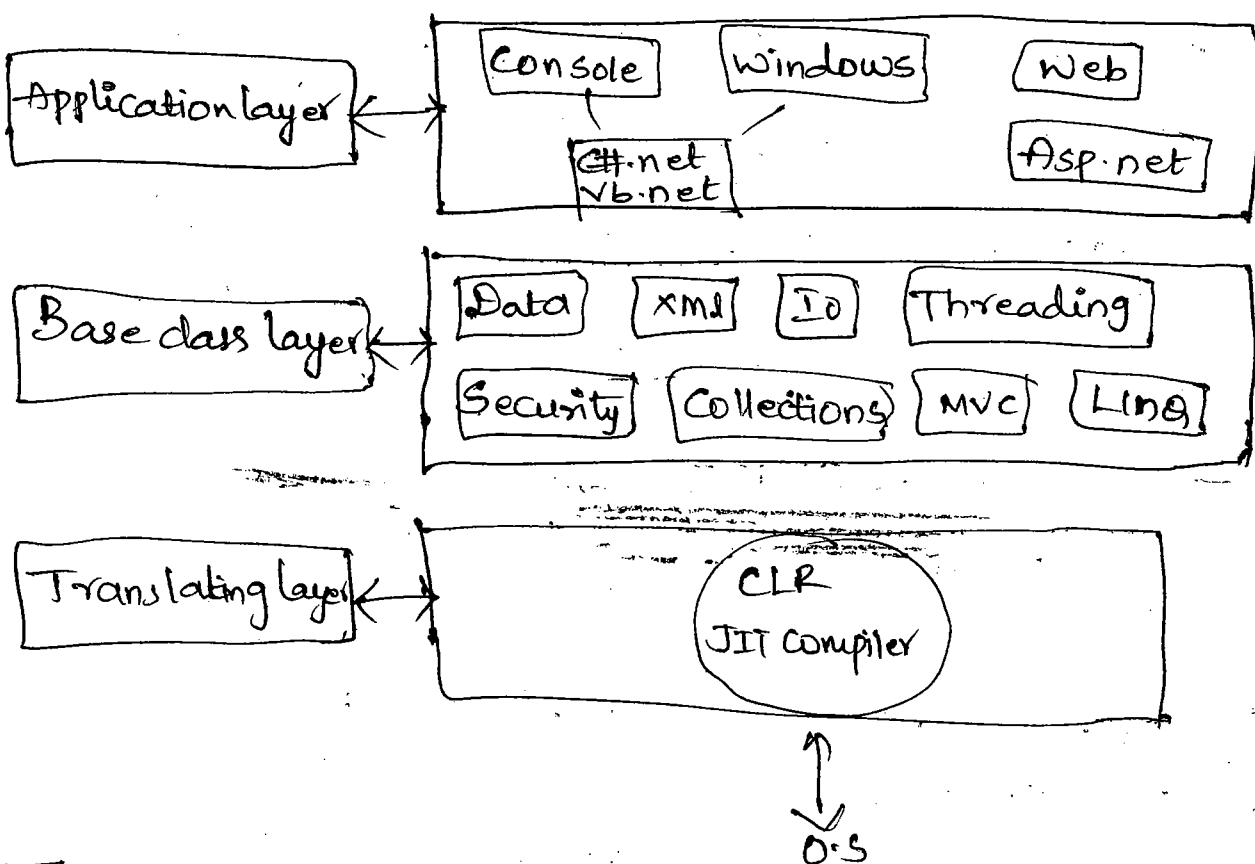
(F) and (T)

(F)

Note:- ~~and~~ && operator will return true if all the conditions are true.

|| operator will return if either of the condition is true.

Architecture of .net framework :-

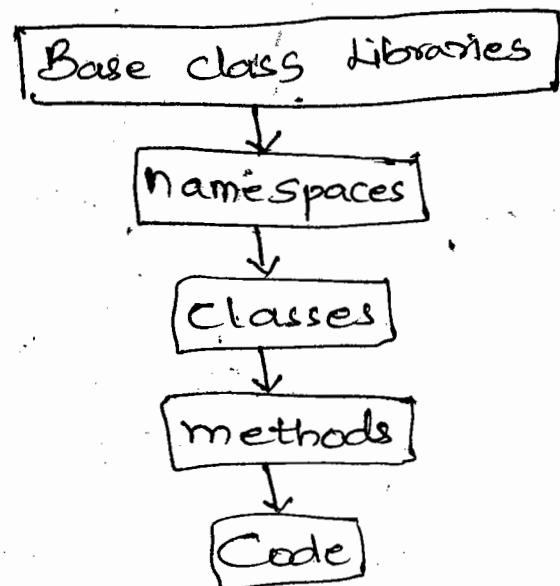


The Architecture of .net framework consists of 3 layers. They are

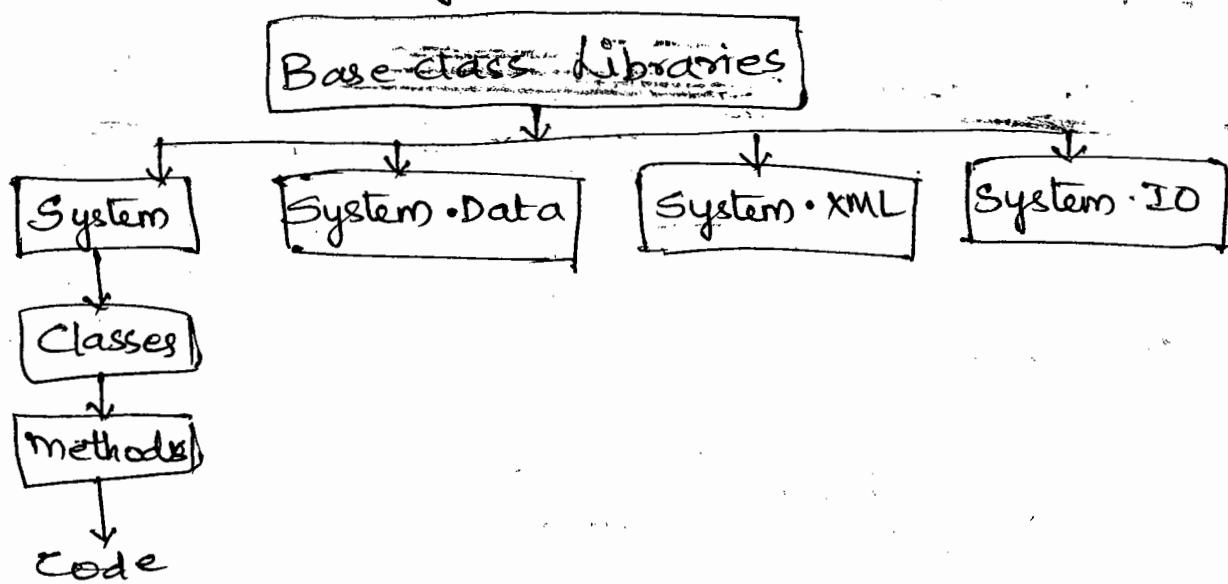
1. Application layer
2. Base class layer
3. Translating layer

Application layer:- This layer is used to design the applications like console , windows forms and web applications by using different types of languages and technologies. like C# .net, VB .net, ASP .net

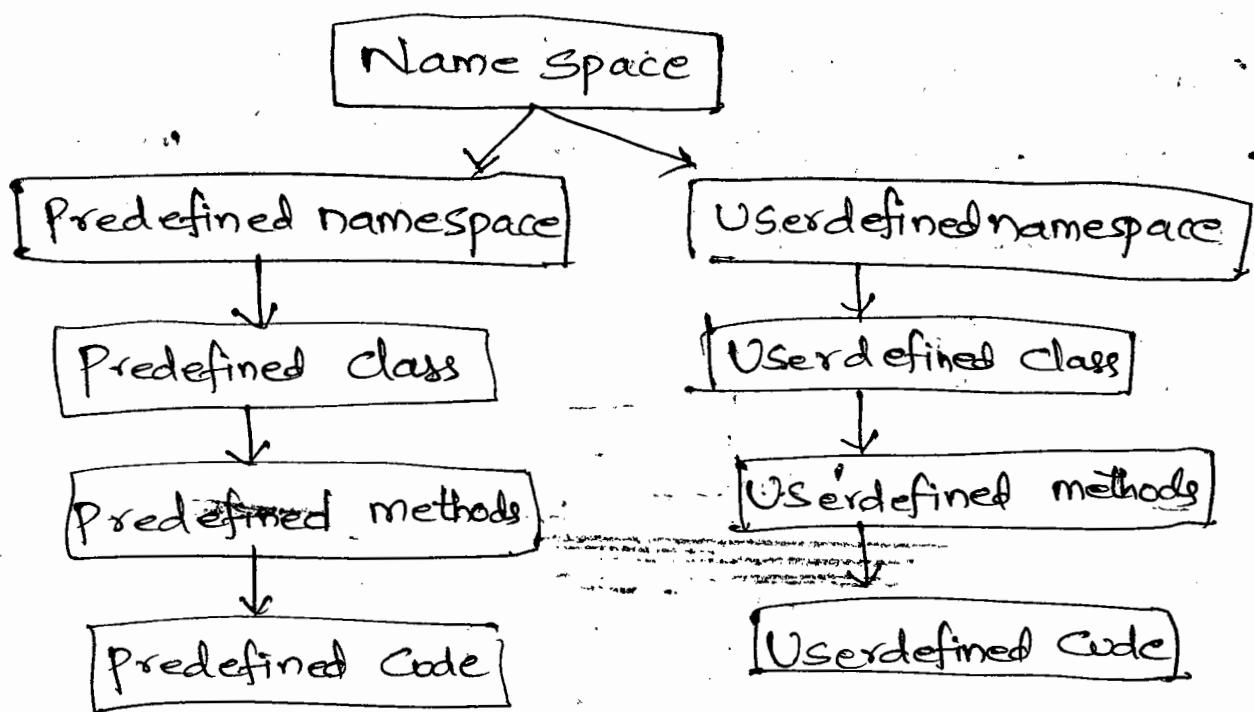
Base class Libraries :- These are used to provide some set of predefined namespaces which consists of some predefined classes which in turn consists of some methods and code.



* Microsoft has provided some set of predefined namespaces, classes and methods in order to write the coding.



NameSpaces are usually divided into two types.



What is Name Space?

Name Space is collection of classes.

Predefined namespace :- The namespaces that was written by Microsoft are called as predefined namespaces.

Userdefined namespace :- The namespace that was created by developer depending on the user requirement is called as userdefined namespace.

C language - Header file

Java - Packages

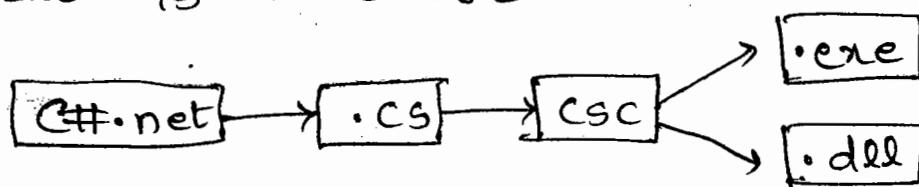
.Net - Namespaces

11th June 2015

What is an Assembly?

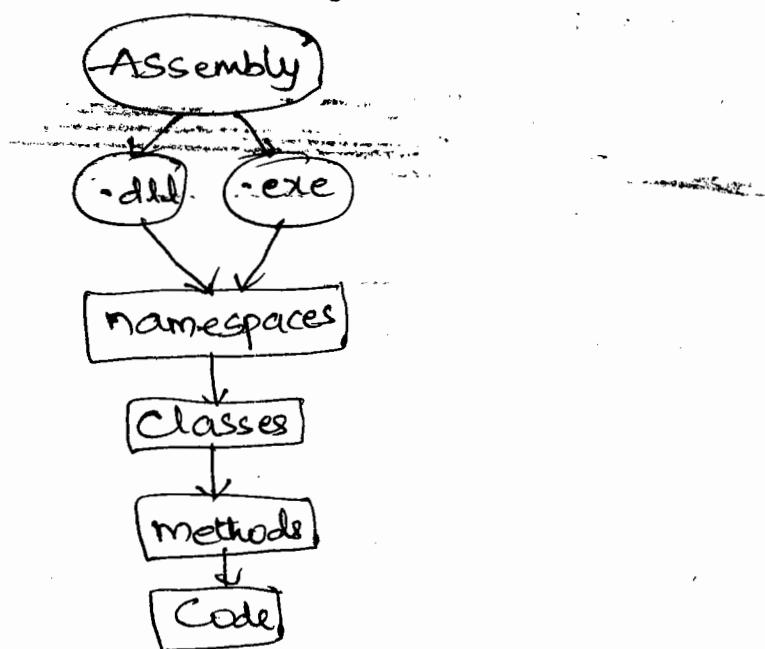
An assembly is compiled format of any .NET program which may be .dll or .exe

- .dll :- Dynamic link library
- .dll is reusable but not executable.
- .exe is Executable but not reusable.



Whenever we compile console app's (or) windows forms app's then the compiler will generate .exe file.

Whenever we compile web application (or) class library then the compiler will generate .dll file.



What is the difference between Assembly and NameSpace?

Assembly is collection of namespaces and namespace is collection of classes.

Syntax for declaring namespace :-

namespace namespace name

{

 class class name

{

}

}

Eg:- for C# net program

namespace System

{

 class Console

{

 public static void WriteLine()

{

}

 public static void Readline()

{

}

}

CLR (Common Language Runtime):-

CLR is the heart and soul of .NET framework.

Architecture of Assembly:-

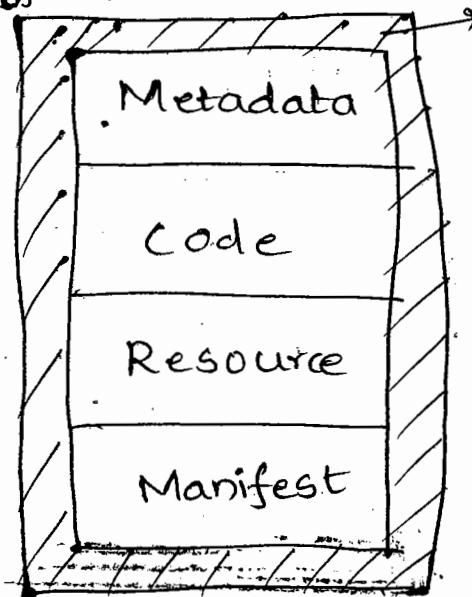
Assembly is also called as MSIL (or) CIL (or) IL

MSIL :- Microsoft Intermediate language

CIL :- Common Intermediate language

IL :- Intermediate language.

Architecture of MSIL:-



Metadata :- The data about data is called

metadata. Metadata consists of the class names and method names.

Code :- Code consists of the actual code.

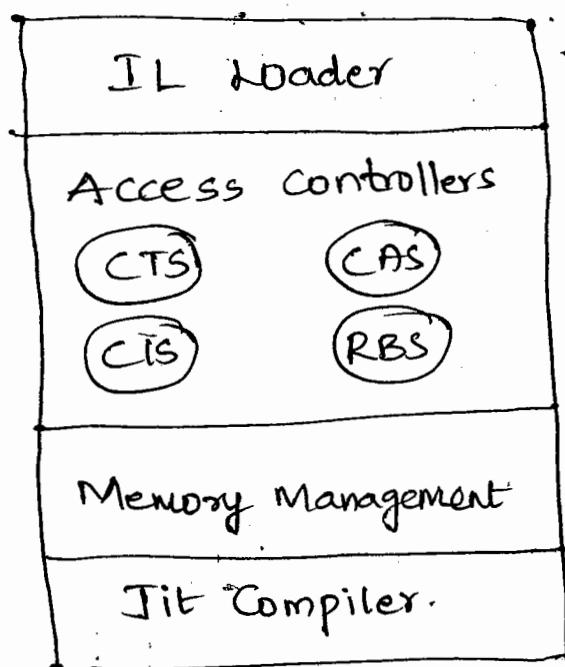
Resource :- Resource consists of images, icons, advertisements etc.

Manifest :- Manifest consists of the Company Information, copyrights.

PE Wrapper :- Portable Executable wrapper class It consists of the ~~take~~ entire information of the IL file.

12th June 2015

Architecture of CLR



- * CLR is the heart and soul of .net framework.
- * CLR (Common Language Runtime) which will take care about memory management, compilation, debugging, type security and converts the code into native code.

IL Loader (Intermediate Language Loader) :-

It Loader is used to load the entire IL file into CLR.

CTS (Common type System) :-

CTS provides common datatypes for all the .NET supportable languages.

CLS (Common language specifications) :-

It is used to provide some set of rules that we have to follow at the time of writing the program.

* CLS provides some set of specifications that has to be followed by different programming languages.

CAS (Code access Security) :-

It is used to check whether the code that is running by CLR is valid code (Or) Invalid code.

RBS (Role Based security) :-

It is used to provide permission check whether the user who is ^(Developing) working with the program is valid or invalid.

Memory Management:-

- CLR will take care about Memory Management.
- ⇒ Memory management means memory allocation and Memory deallocation.
- ⇒ Within the CLR, garbage collector will take care about Memory Management.

JIT Compiler (Just-in-time Compiler):-

It is used to convert MSIL code into native code and finally O.S will communicate with hardware and get the output.

Platform = O.S + Processor

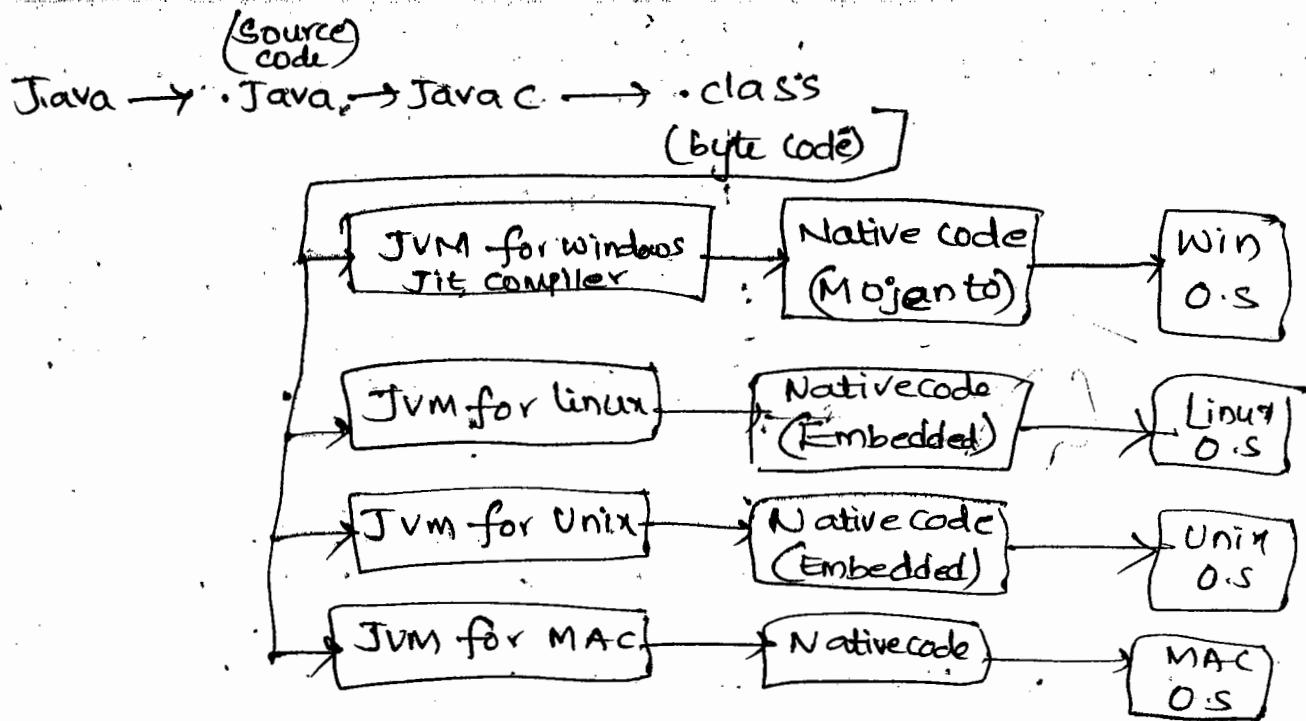
15th June 2015

What is platform dependent?

An application that runs was developed in one operating system can run on the same O.S then that application is called as platform dependent application.

What is platform Independent?

The application that was developed in one operating system can run on any other O.S then that application is called as platform independent app.



* Java is platform independent but jvm is platform dependent.

* The program (Java) that was developed in one operating system can run on any other O.S.

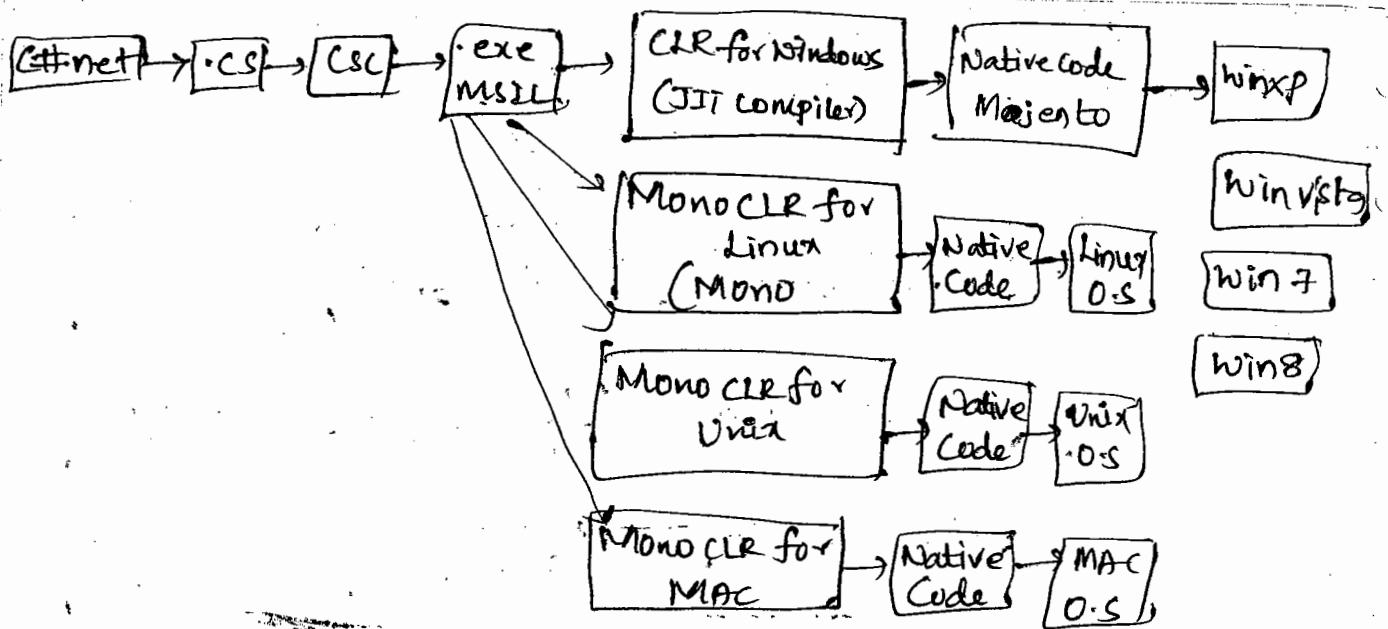
WORA - Write Once Run Anywhere (slogan for Java).

* Different Jvm's were developed for different O.S.

'Net is platform dependent (or) Independent?

• Net is platform dependent with .Net framework and independent with monoframework.

Independent → [Compile in one O.S
Run in another O.S]



Recently, in the year 2015 Microsoft has announced .Net as open source with .Net framework i.e the application that was developed in one O.S can run on any other O.S.

Data types:-

16th June 2015

Data:- Anything that we give as input through Keyboard is called as Data.

Eg:- 10 23 and 20,000

* Data is meaningless.

Information:- Information is the processed data.

* Information is meaningful Data.

Eg:- Anil Salary is 20000

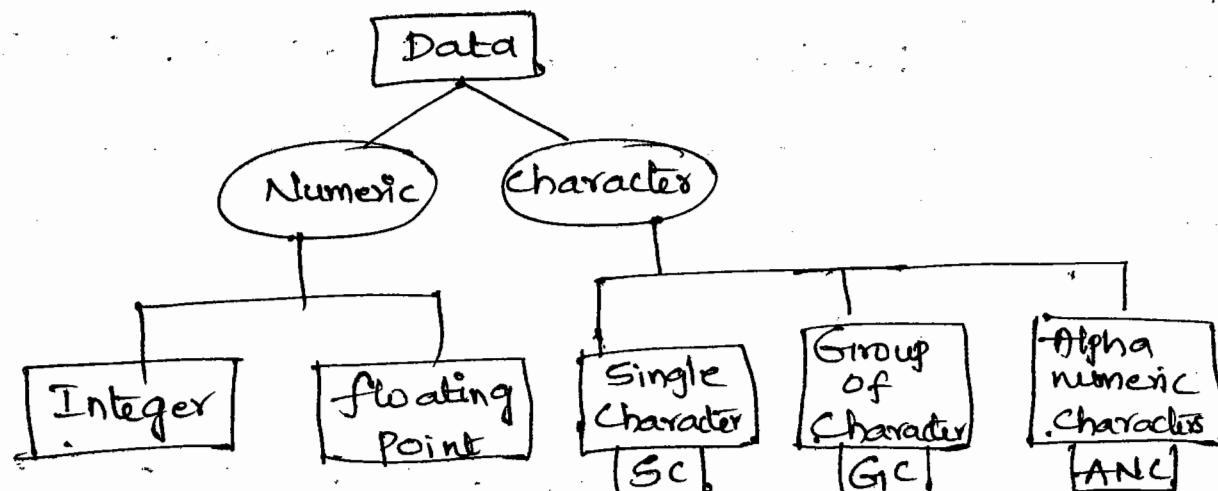
Anil Age is 23

* In any programming languages like C, C++, Java, C# .net, VB.net etc we will work with the Data but not with Information because of 2 reasons.

⇒ Data will occupy less memory compare with information

⇒ It is easier to identify data.

Data is divided into two types.



e.g. 10
25
31
20000

e.g. 2.3
5.6
15.63

e.g. a
b
c

e.g. Anil
Sathya

e.g. 11C2TA0484
TS09A1234
Sravan5051.
@gmail.com.

17th June 2015 → NO NOTES

1 byte(1) 1 float(4)

SC

GIC ANC

1 short(2) 1 double(8)

char(2)

string

1 int(4) 1 decimal(16)

1 long(8)

18th June 2015

What is Datatype?

Datatype specifies the type of data that we store in the memory.

Where the memory is allocated for data?

The memory for the data will be allocated on RAM.

What is byte?

Byte is a datatype which will allocate 1 byte of memory to store integer data.

e.g:- `byte b=10;`

The range of byte is 0 to 255

What is Short?

Short is a datatype which will allocate 2 bytes of memory to store integer data.

e.g:- `short s=25000;`

The range of short is -32768 to +32767

What is Int?

Int is a datatype which will allocate 4 bytes of memory to store integer data.

e.g:- `int i=20;`

The default datatype to store integer value is int.

The max range of int is 10 digit number starting with '2'.

What is long?

long is a datatype which will allocate 8 bytes of memory to store integer data.

eg:- long debit card no = 234689103463

The range of long is 19 digit number starting with '9'.

Float :- float is a datatype which will allocate 4 bytes of memory to store floating point (decimal data).

float a = 3.5; X not valid

float a = 3.5f; ✓ valid.

Double :- Double is a datatype which will allocate 8 bytes of memory to store floating point data
⇒ The default datatype of floating point is double.

eg:- double d = 2.3;

Decimal :- Decimal is a datatype which will allocate 16 bytes of memory to store floating point.

eg:- decimal d = 25736789.567 ; X not valid

decimal d = 25736789.567 M ; ✓ valid.

char:- char is a datatype which will allocate 2 bytes of memory to store a single character.

⇒ char value must be always declared within single quotes [' ']

e.g. char ch = 'a';

String:- string is a datatype which will allocate memory to store group of characters or alpha numeric characters.

⇒ String data must be stored in double quotes [" "]

e.g. String Sname = "anil";

String Panno = "BAGI123456";

* In order to perform operations on data we have to store the data in memory and datatype will allocate memory for data in RAM.

Syntax for declaring datatype :-

Datatype VariableName = Value ;

int . i = 10 ;

int → Datatype

i → Variable

10 → Value (or) data

= → Assignment Operator (AO)

; → end of statement (eos)

Variable :- Variable is an identifier

Variable is the name given for a particular memory location where the data is located.

What is the purpose of Variable ?

The purpose of variable is to identify the data or to access the data.

19th June 2015

Syntax for declaring Variable :-

int i ; // Declaring a Variable with name ;

When we declare the above statement , int is the datatype which will allocate 4 bytes of memory and the name given for that memory is 'i' .

`i=10; // Assigning the value 10 in the memory
whose name is i.`

`int i=10;`

In the above statement at the time of declaring the variable only we are assigning the value. This is called as initialization.

`int i j; ✗ Not valid`

`int i, j; ✓ Valid`

`int i; j; ✗ Not valid`

`int i; int j; ✓ Valid`

`int i, int j; ✗ not valid.`

`int + int = int`

`int * int = int`

`int / int = int`

`int % int = int`

`int + double = double`

`int + float = float`

`double + double = double`

`int + double = double`

`String + String = String`

`any datatype + String = String.`

⇒ + Operator will perform addition between two integers and concatenation operation between two strings.

eg:- $10 + 20 = 30$

"10" + "20" = "1020"

* A variable can store only one value.

eg:- `int i=10,20;` ✗ not valid.

`int i=5;`

`i=7;`

`Console.WriteLine(i);`

Output = 7

`int i=10;`

↓ ↓
Variable Value

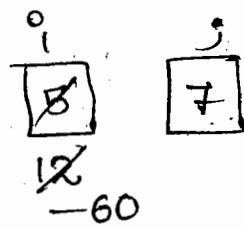
i
10

`int J=i;`

↓ ↓
Variable Value

J
10

int i=5;
 int j=7;
 $i = i + j;$
 C-WL(i); → 12
 $j = j - i;$
 C-WL(j); → -5
 $i = i - i + j * i;$
 C-WL(i); → -60
 $j = j - j + j - j;$
 C-WL(j) → 0



$$i = i + j$$

$$i = 5 + 7 = 12$$

$$j = j - i$$

$$j = 7 - 12 = -5$$

$$i = 12 - 12 + (-5) * 12$$

$$= 12 - 12 - 5 * 12$$

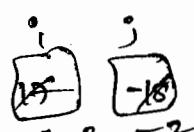
$$= 12 - 12 - 60$$

$$i = -60$$

$$j = 7 - 7 + 7 - 7$$

$$= 0$$

int i = $2 + 3 * 5 - \frac{6}{3.0} + 2;$
 int j = $2 - 3 * 6 + i - 2 - i;$
 C-WL(i); → 12
 C-WL(j); → -18
 $i = j - i + i * 2;$
 C-WL(i); → -1
 $j = i + j - j + i;$
 C-WL(j); → -2
 $i = i + j;$
 C-WL(j); → -3
 C-WL(j); → -2 1



$$j = 2 - 18 + i - 2 - i;$$

$$= 2 - 18 + 12 - 0 - 2 - 12$$

$$= -16 + 12 - 0 - 2 - 12$$

$$= 1 - 2 - 12 - 0$$

$$= -1 - 12$$

$$= -18$$

$$= -1$$

$$j = -1 + (-18) - (-18) + (-1)$$

$$= -1 - 18 + 18 - 1$$

$$= -2$$

$$j = -2 - (-3)$$

$$= -2 + 3$$

$$= 1$$

$$\begin{array}{l} i = i + j \\ = -1 - 2 \\ = -3 \end{array}$$

20th June 2015

$$\text{int } i = \frac{5}{2} + 3 - 5 * \frac{6}{11} - 7 * 6;$$

$$\text{int } j = 3 - 2 * 6 + 7 - 3;$$

$$\text{bool } b = i > j;$$

$$CWL(b);$$

$$\text{bool } b_1 = i == j;$$

$$CWL(b_1);$$

$$\text{bool } b_2 = i > j \text{ } \& \& \text{ } i \leq j;$$

$$CWL(b_2);$$

$$\text{bool } b_3 = i \leq j \text{ } \& \& \text{ } i == j \text{ } \& \& \text{ } i != j;$$

$$CWL(b_3);$$

$$CWL(i); \rightarrow -89$$

$$CWL(j); \rightarrow -5$$

$$CWL(b); \rightarrow \text{false}$$

$$CWL(b_1); \rightarrow \text{false}$$

$$CWL(b_2); \rightarrow \text{false}$$

$$CWL(b_3); \rightarrow \text{true}$$

$$i = 2 + 3 - \frac{30}{11} - 42$$

$$= 2 + 3 - 2 - 42$$

$$= -39 - 39$$

$$-39 > -5 \quad -39 == -5$$

(false)

$$j = 3 - 12 + 7 - 3$$

$$= -9 + 7 - 3$$

$$-2 - 3 = -5$$

$$17 \frac{3}{2} \in \mathbb{Z}$$

$$-39 > -5 \text{ } \& \& \text{ } -39 \leq -5$$

false

true

(false)

$$-39 \leq -5 \text{ } \& \& \text{ } -39 == -5 \text{ } \& \& \text{ } -39 != -5$$

(true) (false)

(true)

(true)

$$\text{int } i = 2 * 3 + 2 - \frac{5}{2} - \frac{7}{11}$$

$$\text{int } j = 7 - 3 * 3 + \frac{7}{9} * \frac{7}{9}$$

$$CWL(i);$$

$$CWL(j);$$

$$CWL(i+j);$$

$$\text{bool } b = i > j;$$

$$CWL(b);$$

$$\text{bool } b_1 = i \leq j;$$

$$CWL(b_1);$$

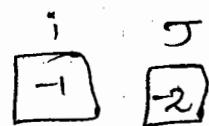
$$b_1 = i == j;$$

$$CWL(b_1);$$

$$i = 6 + 2 - 2 - 7 \quad J = 7 - 9 + 0 * 7$$

$$= -1 \quad \quad \quad = 7 - 9$$

$$= -2$$



$$(i+j) = -1 + -2 = -3$$

$$-1 > -2 \text{ (T)}$$

$$-1 \leq -2 \text{ (F)}$$

$$-1 == -2 \text{ (F)}$$

\Rightarrow byte a=10;

byte b=20;

byte c=a+b;

Console. WriteLine(c);

What is the output of the program?

When we execute the above program it displays the error message.

* /wap to swap two numbers */

{ int i=5;

int j=6;

int k;

Console. WriteLine("before swap i is" + i); $\rightarrow 5$

Console. WriteLine (" before swap j is" + j); $\rightarrow 6$

K=i;

i=j;

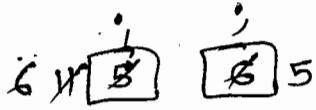
j=k;

Console. WriteLine("After swap i is" + i); $\rightarrow 6$

Console. WriteLine("After swap j is" + j); $\rightarrow 5$



* Swap two numbers without third variable / *

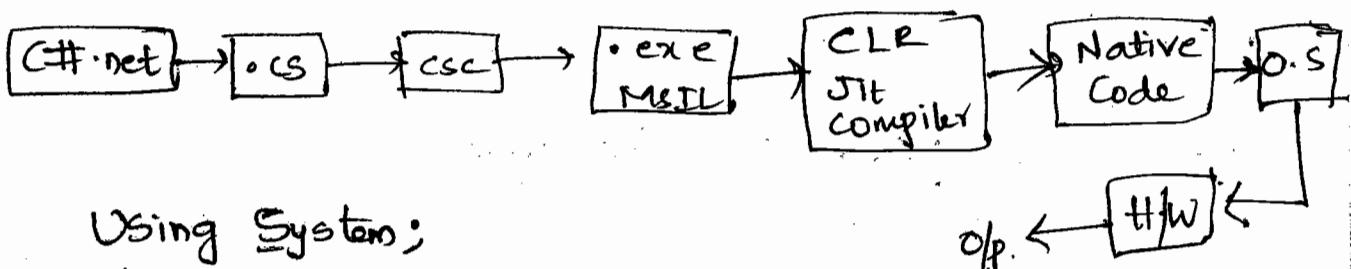


$$i = i + j ; \rightarrow 11 \quad (5+6=11)$$

$$j = i - j ; \rightarrow 5 \quad (11-6=5)$$

$$i = i - j ; \rightarrow 6 \quad (11-5=6)$$

22nd June 2015



```
Using System;
Class Classname
{
    Static Void Main()
    {
        Console.WriteLine("Welcome to C#-net");
    }
}
```

System → S Capital

Classname → C Capital

Main → M Capital

Console → C Capital

Write → W Capital

Line → L Capital

Steps to write the first program in C# .net:-

Step1:- Go to 'E' Drive and create a folder (Directory) with name Demoprograms.

Step2:- Open notepad and write the program

Using System; (S Capital)

Class Firstprogram (F Capital)

{

 Static void Main() (M Capital)

{

 Console. ~~Write~~ ~~Writeline~~ → (C, W, L Capitals).

 Console.WriteLine("Welcome to c#.net");

}

}

Step 3:- Save the program in E:\Demoprograms

Firstprogram.cs

Step 4:- Compile the program

* Microsoft has given a Command prompt which is called as visual studio command prompt to compile and execute the program.

* Visual studio Command prompt will be installed automatically when we install visual studio editor.

Step 5:- go to → Start → Programs → Microsoft Visual Studio-2010 → Visual Studio Tools → Visual Studio Command Prompt.

Step 6:- Change the drive
E: and press enter.

Step 7:- change the Directory

Syntax: cd directory name

Eg:- cd Demoprograms

and press enter

Step 8:- compile the program.

CSC filename.cs

Eg:- CSC Firstprogram.cs and press enter

Whenever we Compile the program the compiler will check for Syntax errors if there are no Syntax errors then CSC compiler will generate Firstprogram.exe file.

→ go and check in E:\Demoprograms

Step 9:- Execute the program.

Firstprogram.exe Type " in Command Prompt.

→ Whenever we execute the program within the CLR, Jit compiler will convert MSIL code to native code and given to O.S. Finally O.S. will communicate with hardware to get the output.

* WAP to perform addition of two numbers *

Using System;

Class AddDemo

{

Static Void Main()

{

int i=10;

int j=20;

int K=i+j;

Console.WriteLine ("Sum is "+K);

}

}

O/p :- Sum is 30.

23rd June 2015

* WAP to declare firstname and lastname and display the full name */

Using System;

Class Fullname demo

{

Static Void Main()

{

String fname="sathya";

String lname="technologies";

String fullname=fname + lname;

```
Console.WriteLine ("full name is" + fullname);
```

```
}
```

```
}.
```

O/p :- fullname is Satya Technologies

Q/WAP to declare Student no, S.name, M1, M2, M3 and calculate total marks and % of marks and display the student details.

using System;

```
class Totalmarks demo5
```

```
{
```

```
static void main()
```

```
{
```

```
String S.no = "11C2IA0484";
```

```
String S.name = "Sravankumar";
```

```
int m1 = 95;
```

```
int m2 = 95;
```

```
int m3 = 90;
```

```
int total = m1 + m2 + m3;
```

```
Console.WriteLine
```

```
("Percentage is" + Percentage);
```

```
}
```

```
}
```

$$\frac{95+95+90}{3} = \frac{280}{3}$$

double Percentage = $\frac{m_1+m_2+m_3}{Total}$;
double Percentage = $\frac{Total}{3*0}$;

```
Console.WriteLine ("Student no is" + S.no);
```

```
Console.WriteLine ("Student name is" + S.name);
```

```
Console.WriteLine ("Marks 1 is" + m1);
```

```
" " " ("Marks 2 is" + m2);
```

```
" " " ("Marks 3 is" + m3);
```

```
" " " ("Total marks is" + total);
```

What is local variable?

The Variable that was declared within the method is called as local variable.

⇒ Local Variable must be initialised.

⇒ Local Variable must be used in the program otherwise the compiler will display the warning

saying that "Variable is declared but it was never used".

#WAP to declare book I.D, book name , Quantity , Price calculate total bill and display the output.

using System;

class Totalbill demo

{

Static Void Main()

{

int book I.D = 1;

String bookname = " C#-net";

int Quantity = 2;

double price = 200.50;

double Total bill = Quantity * price;

Console.WriteLine ("book I.D is" + book I.D);

" " ("bookname is" + bookname);

" " ("Quantity is" + Quantity);

" " ("Price is" + Price);

" " ("Total bill is" + Total bill);

* WAP to declare employee no., e.name, basic salary.
Calculate D.A (Dearest Allowance), H.R.A (House Rental Allowance)
and total salary and display them.

using System;

Class Employee

{

Static Void Main()

{

int e.no = 001;

String e.name = "Sravankumar";

double b.sal = 20000.50;

double d.a = $\frac{20}{100} * b.sal$; $(0.2 * b.sal)$

double h.ra = $\frac{40}{100} * b.sal$; $(0.4 * b.sal)$

double t.sal = b.sal + d.a + h.ra;

Console.WriteLine("e.no is" + e.no);

" " ("e.name is" + e.name);

" " ("d.a is" + d.a);

" " ("h.ra is" + h.ra);

" " ("t.sal is" + t.sal));

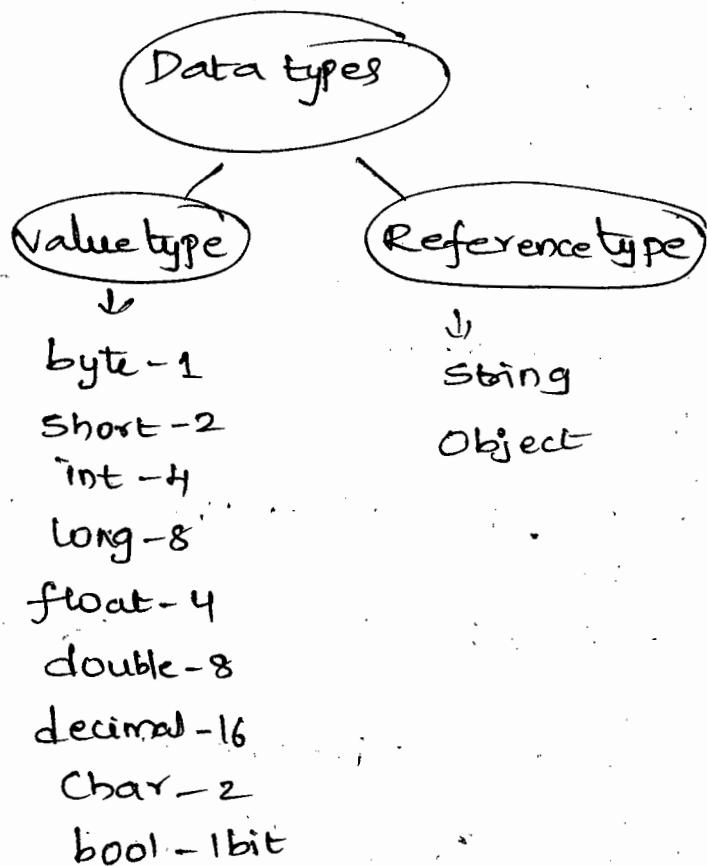
}

}

24th June 2015

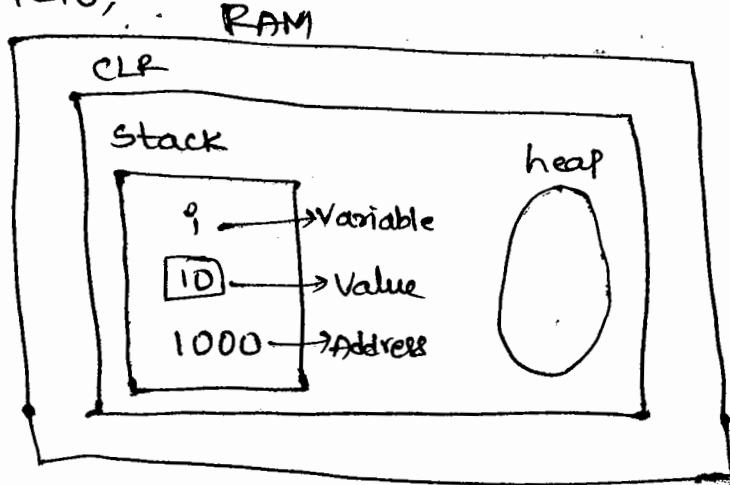
Data types are divided into two types:

1. Value type Data type
2. Reference type Data type.



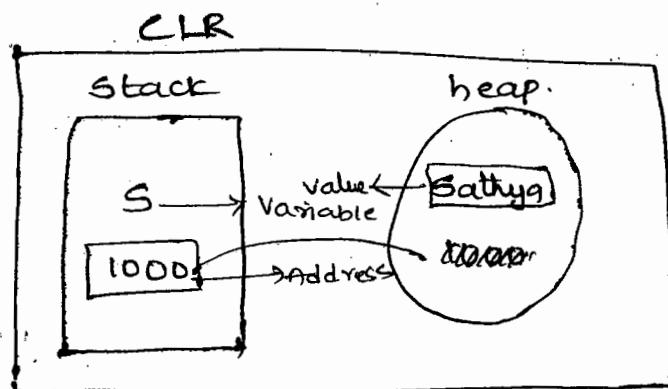
Value type Datatype :- The datatype whose value, variable and the address of the value will be stored on stack memory is called as Valuetype datatype.

e.g:- int i=10;



Reference type data type :- The data type whose value will be stored on heap and the address of the value will be stored on stack memory along with the variable name is called as Reference type datatype.

e.g:- String S = "Sathy";



Type Casting (Type Conversion) :- It is the process of converting one data type value to the another data type.

Type casting: Casting is of two types.

- * Implicit type casting.
- * Explicit type casting.

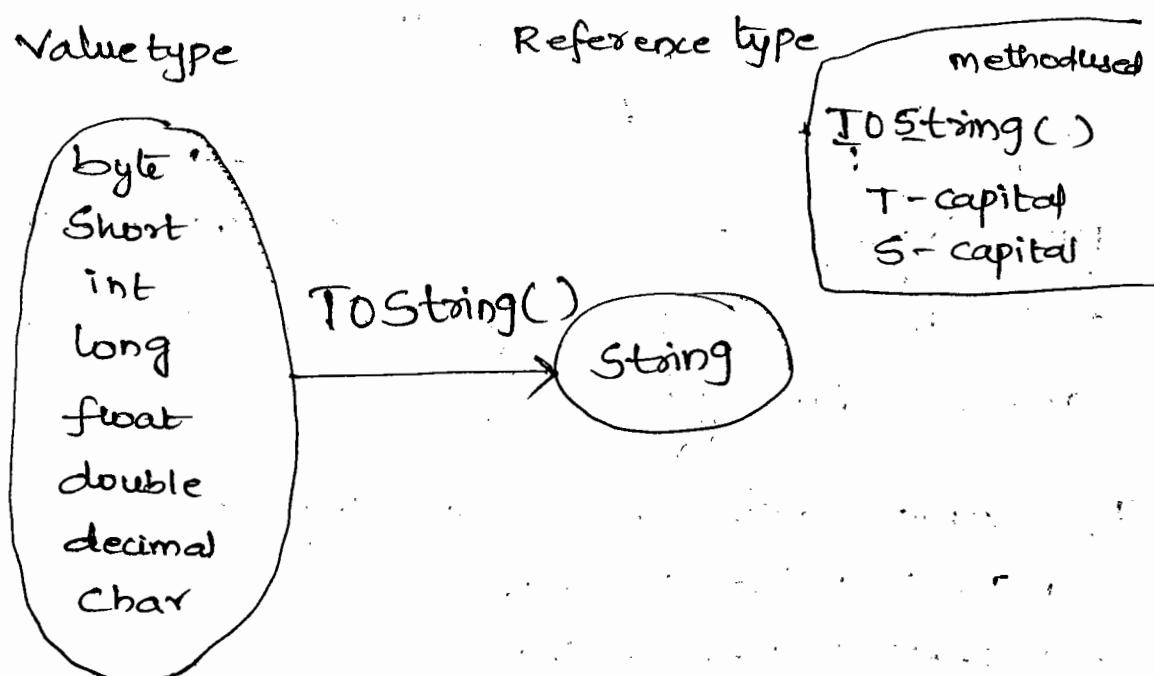
Implicit type Casting :- It is not required to write separate code to convert from one datatype to another datatype with implicit casting

e.g:- whenever we convert smaller datatype value to longer datatype, it is not required to write a separate code to convert.

Explicit type casting :- While working with explicit type casting we need to write a separate code to convert from one datatype to another datatype.

Boxing :- Boxing is a process of Converting value type datatype to reference type datatype.

* Boxing requires explicit type casting.



e.g.:- for Boxing

```
using System;
Class Boxingdemo
{
    static void Main()
    {
    }
}
```

```
Console.WriteLine("Value of i is "+i);
Console.WriteLine("Value of s is "+$);
Console.WriteLine(i+$);
```

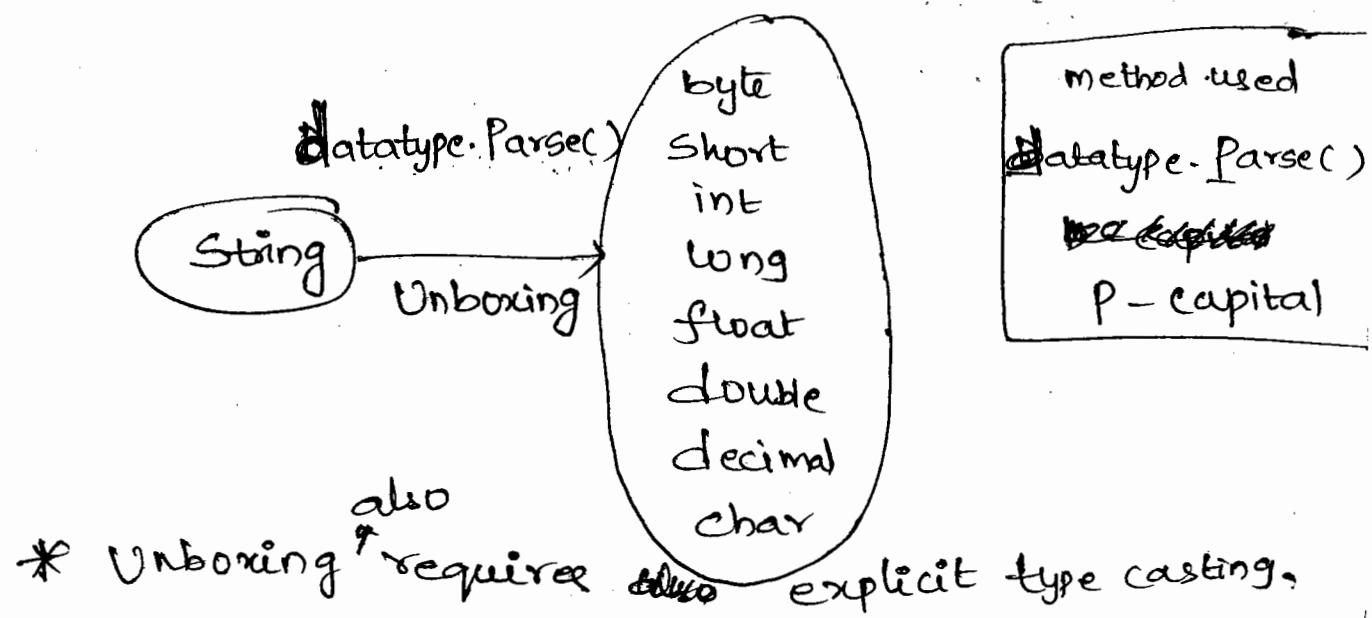
String S = i.ToString()); O/p :- value of i is 10.
Value of S is 10
1010.

** WAP to convert from double to string **

```
using System;
class DtoSdemo{
{
static void Main()
{
    double i = 2.34;
    String s = i.ToString();
    Console.WriteLine("Value of i is " + i);
    " " ("Value of s is " + s);
    " " (i + s);
}
}
```

O/P Value of i is 2.34
Value of s is 2.34
2.342.34

Unboxing :- It is a process of converting reference type datatype to value type datatype.



Eg:-

String $\xrightarrow{\text{int.Parse()}}$ int

String $\xrightarrow{\text{float.Parse()}}$ float

String $\xrightarrow{\text{double.Parse()}}$ double

String $\xrightarrow{\text{char.Parse()}}$ char

Eg:- Using System();
Class Unboxingdemo.

```
{  
    static void Main()  
    {  
        string s = "10";  
        int i = int.Parse(s);  
        Console.WriteLine("Value of s is "+s);  
        " " ("Value of i is "+i);  
    }  
}
```

O/p:- Value of s is 10
Value of i is 10.

~~#include <iostream.h>~~ // NAP to Convert from string to double */

```
using System();
Class Stoddemo
{
```

```
Static Void Main()
{
    String s = "2.34";
    Double d = double.Parse(s);
    Console.WriteLine ("Value of s is "+s);
    "           " . ("Value of d is "+d);
```

```
}
```

```
}
```

O/p:- Value of s is 2.34
Value of d is 2.34.

[25th June 2015]

Widening :- It is a process of converting smaller datatype value to longer datatype.

* Widening Supports implicit typecasting i.e it is not required to write a separate code to convert from one datatype to another data type.

e.g:-

```
using system;
```

```
Class Wideningdemo
```

```
{
```

```
Static Void Main()
```

```
{
```

```
int i=10;
```

```
long l=i;
```

```
Console.WriteLine("Value of i is "+i);
```

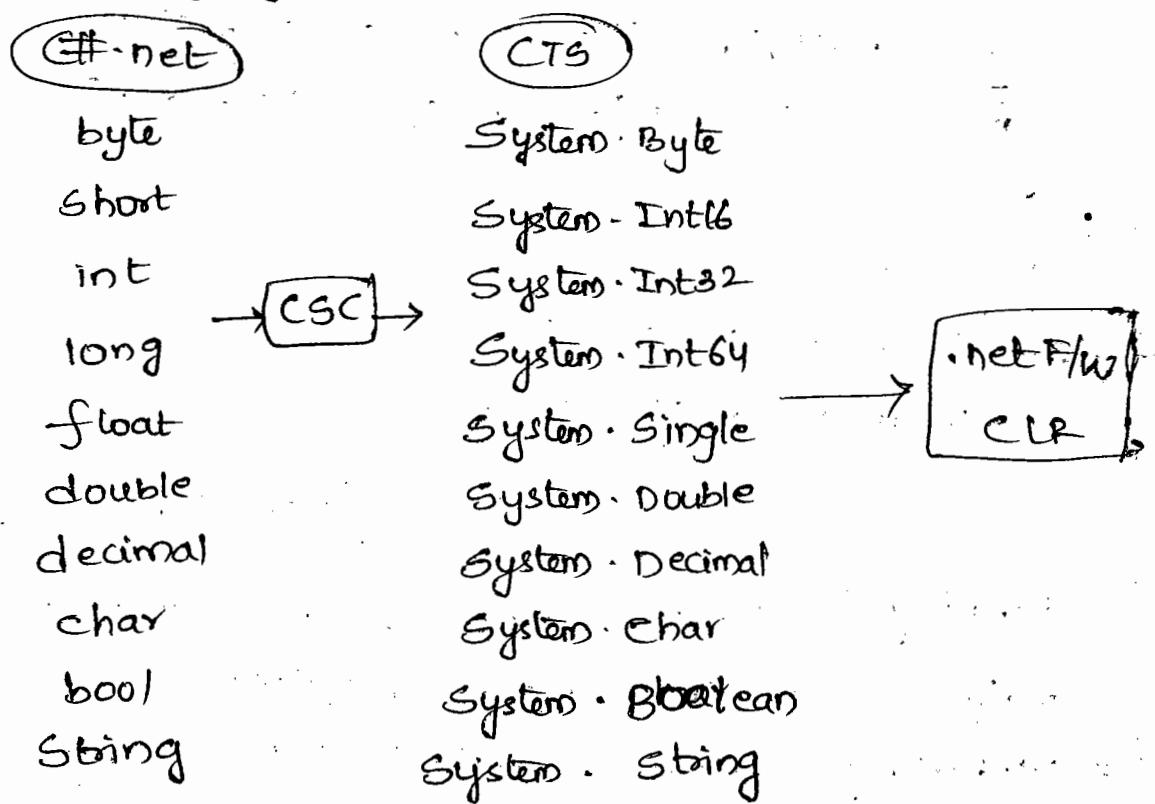
```
Console.WriteLine("Value of l is "+l);
```

Narrowing :- It is a process of converting longer datatype value to smaller datatype value.

* Narrowing requires explicit type casting.

What is CTS? (Common Type System) :-

CTS provides common datatypes for all the .NET supportable languages.



⇒ .NET is a framework that supports multiple languages.

⇒ Different languages will have different datatypes.

⇒ Every language will have their own compilers.

for eg:- C# .NET have CSC compiler

~~it converts as~~

At the time of compilation of the program the compiler will convert C# .NET datatype to CTS type because .NET, f/w, can execute only CTS types.

Any datatype Convert.ToInt32() → int
C - capital
T - capital
I - capital

Any datatype Convert.ToInt64() → long

Any datatype Convert.ToSingle() → float

Any datatype Convert.ToDouble() → Double

eg:- for Narrowing:-

using System;

class Narrowingdemo

{

 static void Main()

{

 long l = 10;

 int i = Convert.ToInt32(l);

 Console.WriteLine("Value of l is " + l);

 Console.WriteLine("Value of i is " + i);

}

Note :- Narrowing is not considered for max. cases.
Widening is mostly considerable.

Note :- Narrowing is not safe type of programming
because whenever we convert longer data type
value to smaller data type then data accuracy
is not maintained.

Console.ReadLine()

It is used to read the input from the console.

⇒ By default Console.ReadLine will read the input in the form of string.

WAP to accept firstname and lastname from console and display the fullname.*/

```
using System;
class Fullnamedemo
{
    static void Main()
    {
        Console.WriteLine("Enter First name");
        string fname = Console.ReadLine();
        Console.WriteLine("Enter last name");
        string lname = Console.ReadLine();
        Console.WriteLine(fname + lname);
    }
}
```

Op:- Enter first name

Sathya

Enter last name

technologies

Sathya technologies.

/* A way to accept two numbers from console and perform addition operation */

Using System;

Class Additiondemo

{

Static Void Main()

{

Console.WriteLine ("Enter First no");

int a = int.Parse (Console.ReadLine());

Console.WriteLine ("Enter Second no");

int b = int.Parse (Console.ReadLine());

int sum = a+b

Console.WriteLine ("Sum is "+sum);

}

}

Output :-

Enter first no

10

Enter second no

20

Sum is 30.

26th June 2015

Conditional Statements:-

The statements that will get executed based on the conditions are called as Conditional Statements.

- * In order to check the condition, we have to use Conditional Operators like $<$, $>$, \leq , \geq , \neq , $=$
- * Conditional operator will always return a boolean value either true or false.

Syntax for if :-

```
if (condition) {  
}  
} → Block.  
}
```

What is Block?-

The code that was written within the flower braces is called Block.

* If Aspects When to declare a block?

If we want to execute some set of statements when the condition is true or false, then we have to declare the statements within the block.

* In Asp.net we can perform validations by using only if.

```
if (10 >= 5)  
    C.WL ("Hello");
```

→ only one statement so no flower braces.

```
if (10 <= 10)  
{  
    C.WL ("Hello");  
    C.WL ("Hi");  
}
```

→ There are two statements, so we use flower braces.

If - else :- Syntax:-

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

* If the condition is true then the code that was written within if will get executed.

* If the condition is false then the code that was written within the else part get executed.

10

60

e.g:- if ($2+3-5+2*3+5 \geq 25-5+40$). (F)

C.WL ("Hello");

else

C.WL ("Hi");

O/P:- Hi

e.g:- if ($-5+20-49 \leq -25$)

{

C.WL ("Sathya");

C.WL ("Technologies");

}

else

{

C.WL ("C#-net");

C.WL ("Asp.net");

}

O/P:- Sathya
Technologies.

* WAP to accept two numbers from console
if the sum of the numbers is ≥ 0 , then
print positive number else print negative
number.

using System;

Class Demo;

{

Static void Main()

{

C.WL ("Enter first no");

int a = int.Parse (Console.ReadLine);

C.WL ("Enter second no");

int b = int.Parse (Console.ReadLine);

int sum = a+b;

if (sum ≥ 0)

C.WL ("Positive sum");

else

C.WL ("negative sum");

}

}

O/P:- Enter first no

10

Enter second no

30

Positive sum.

Program to accept a number and check whether the given number is even number or odd number

```
{  
    C.WL ("Enter a number");  
    int a = int.parse (Console.ReadLine());  
    if (a%2 == 0)  
        C.WL ("Given number is even number");  
    else  
        C.WL ("Given number is odd number");  
}  
}
```

Else-If :- Syntax

```
if (      )  
{  
}  
else if (      )  
{  
}  
else if (      )  
{  
}
```

If we want to execute only one condition among group of conditions then we have to go for else if.

Write a program to accept two numbers from console and check which no. is greater if

{

C.WL ("Enter first number");

int a = int.parse (console.ReadLine());

C.WL ("Enter second number");

int b = int.parse (console.ReadLine());

if (a > b)

C.WL ("a is greater");

elseif (a < b)

C.WL ("b is greater");

else if (a == b)

C.WL ("Both are ^{equal} greater");

}

}

Nested If :- Syntax

If (Condition)

{

If (Condition)

}

WAP to accept student no, S.name, m₁, m₂, m₃
Calculate total marks & percentage of marks if
 $\text{Per} \geq 75$ and $\text{Per} \leq 100$ then print first class
if $\text{Per} \geq 60$ and $\text{Per} \leq 75$ then print second class
else print fail.

```
{  
Console.WriteLine("Enter Student no");  
int Sno = int.Parse(Console.ReadLine());  
CWL ("Enter student name");  
String Sname = Console.ReadLine();  
CWL ("Enter m1");  
int m1 = int.Parse ("C.RL()");  
CWL ("Enter m2");  
int m2 = int.Parse ("C.RL()");  
CWL ("Enter m3");  
int m3 = int.Parse ("C.RL()");  
int total = m1 + m2 + m3;  
double per =  $\frac{\text{total}}{3.0}$ ;  
if (per  $\geq 75$  && per  $\leq 100$ )  
CWL ("First class");  
else else if (per  $\geq 60$  && per  $\leq 75$ )  
CWL ("Second class");  
else  
CWL ("Fail");  
}  
}  
}
```

Different ways of writing program. (27th June 2015)

if ($a > b \&& a > c$) a is big else if ($b > a \&& b > c$) b is big else if ($c > a \&& c > b$) c is big	if ($a > b$) if ($a > c$) a is big else if ($b > c$) b is big else c is big	if ($a > b \&& a > c$) a is big else if ($b > c$) b is big else c is big elseif ($a == b \&& a == c$, $b == c$) No two numbers must be equal.
--	---	---

What is the difference b/w size of() and type of()?

Size of() is used to get the size of the datatype

Console.WriteLine (size of (int));

Type of() is used to get the base datatype name
i.e. its type

Console.WriteLine (type of (int));

System.Int32

Switch - Case:- If we want to check only one condition directly among group of conditions then we have to go for Switch - case.

(Based on the requirement the condition is directly checked the condition instead of checking all the conditions).

Syntax for switch-case:-

Switch (condition)

{

case 1:

break;

Case 2:

break;

default:

break;

}

eg for Switch Case:-

```
C·WL ("enter a no");
int no = int · Parse (C·RLC));
```

Switch (no)

{

Case 1:

C·WL (One);

break;

Case 2:

TWO;

break;

Case 3:

three;

break;

break - if
default - else

Case 4:
four;
break;
default:
enter valid no
break.

Eg:- for switch case:-

using System;
Class Switchdemo

{

Static void Main()

{

Console.WriteLine ("Enter a no");

int no = int.Parse (Console.ReadLine());

Switch (no)

{

• Case 1:

Console.WriteLine ("one");

break;

Case 2:

C.WL ("Two");

break;

Case 3:

- C.WL ("Three");

break;

Case 4:

C.WL ("Four");

break;

default :

C.WL ("enter no b/w (-4)");

break; } }

* WAP to accept a character from console and
Check whether the given character is vowel or consonant.

{

C.WL ("Enter a character");

Char ch = Char.parse (C.RL());

Switch (ch)

{

Case 'a':

C.WL ("a is vowel");

break;

Case 'e':

C.WL ("e is vowel");

break;

Case 'i':

C.WL ("i is vowel");

break;

Case 'o':

C.WL ("vowel");

break;

Case 'u':

C.WL ("vowel");

break;

default:

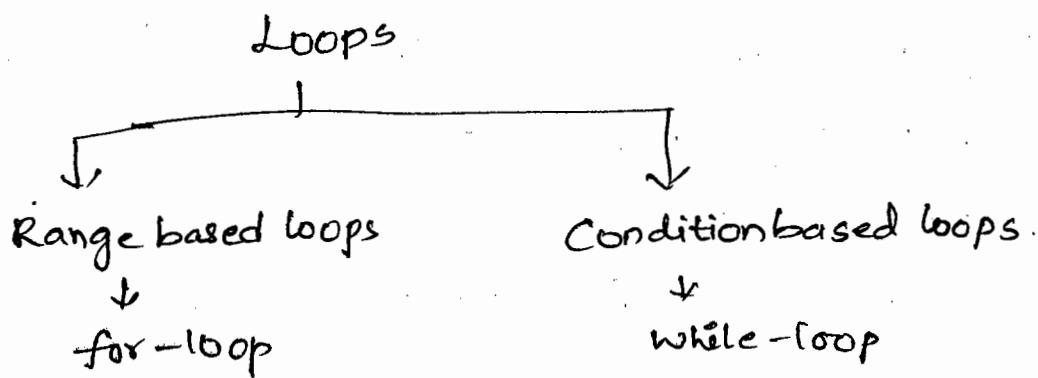
C.WL ("consonant");

break; } }

Loops

28th June 2015

Loop:- Loop is a mechanism which is used to execute some set of statements repeatedly again and again until the condition becomes false.



- * If we know the range we have to go "for-loop".
- * If we don't know the range we have to go "while-loop".

```
int i=10; // Initialization  
i++; // assignment
```

Initialization: It is used to initialize the starting value.

Condition: A condition is used to compare two or more values and it returns a boolean value.

$i \leq 10 \rightarrow \text{Condition}$

Assignment: The values stored after initialization

Increment / Decrement Operators:-

Increment :- $i++ \rightarrow$ incrementing i Value by 1 time

Increment:- $i++$ → incrementing i value by 1 time

$i=i+2$ → incrementing i value by 2 times.

Decrement:- $i--$ → decrementing i value by 1 time

$i=i-2$ → decrementing i value by 2 times.

* Every loop consists of three sections.

* Initialization.

* Condition.

* Increment/Decrement.

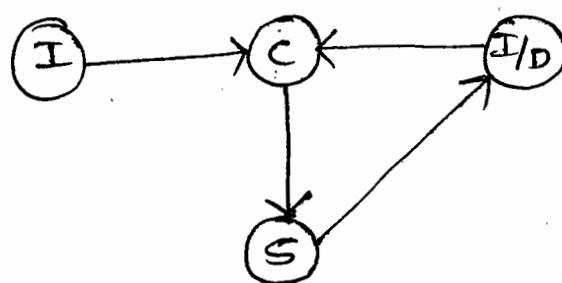
The syntax for for-loop :-

for (initialization; condition; inc dec)

{

}

Execution of for loop :-



Step-1 = Initialization

Step-2 = Condition

Step-3 = Statements

Step-4 = Inc/dec

Step-5 = condition

Step-6 = statements

Step-7 = Inc/dec - so on.

/*WAP to print Sathya technologies 5 times*/

using System;

class Demo

{

static void Main()

{

for(int i=1; i<=5; i++)

CWL("Sathya technologies")

}

}

/*WAP to accept 5 numbers from console and find
the sum of numbers */

{

int Sum=0;

for(int i=1; i<=5; i++)

{

CWL("Enter a no");

int no = int.Parse(C.RL());

Sum = Sum + no;

}

CWL("Sum");

/* WAP to accept 6 numbers from console and find
the sum of even numbers and sum of odd
numbers */

{

```
int evenSum = 0;  
int oddSum = 0;  
for (int i=1, i<=6, i++)  
{  
    C::WL (" enter a no");  
    int no = int::parse (C::RL());  
    if (no % 2 == 0)  
        evenSum = evenSum + no;  
    else  
        oddSum = oddSum + no;  
}  
C::WL (" even sum ");  
C::WL (" odd sum ");  
}
```

* WAP to accept 7 numbers from console and find
the sum of positive numbers and negative numbers.

{

int possum=0;

int negsum=0;

for (int i=1; i<=7; i++)

{

CWL ("enter a no");

int no = int parse (C.RLC());

if (no >= 0)

pos sum = possum + no;

else

neg sum = negsum + no;

}

CWL (possum);

CWL (negsum);

}

/* WAP to accept 8 no's from Console and count
the no. of even no's and no's of odd no's and
display the even count and odd count.

{

```
int evenCount = 0;  
int oddCount = 0;  
for (int i=1 ; i <= 8 ; i++)  
{  
    C.WL ("Enter a no");  
    int no = int.parse (C.RLC);  
    if (no % == 0)  
        evenCount = evenCount + 1;  
    else  
        oddCount = oddCount + 1;
```

}

```
C.WL (evenCount);  
C.WL (oddCount);
```

}

* WAP to accept 9 numbers from Console and
Count the no. of positive no's and no. of negative no's.

{

int posno=0;

int negno=0;

for (int i=1; i <=9; i++)

{

CWL ("enter a no");

int no = int.parse (C.RLC);

if (no >= 0)

posno = posno+1;

else

negno = negno+1;

}

CWL (~~and~~ posno);

CWL (negno);

}

* WAP to find the sum of given no i.e 123456

```
{
```

```
int no = 123456;
```

```
int r = 0;
```

```
int sum = 0;
```

```
for (int i=1; i<=6; i++)
```

```
{
```

```
r = no%10;
```

```
sum = sum + r;
```

```
no = no/10;
```

```
}
```

```
CWL ("sum is" + sum);
```

Note:- In order to identify the units place number we have to use $\boxed{no \% 10}$

* In order to remove the identified no.

use $\boxed{no / 10}$

<u>Execution</u>			
no	r	Sum	i
1234	6	0	1
123	5	6	2
12	3	6	3
1	2	9	4
	1	10	

$i=1$ $1234 \% 10 = 4$ $sum = 4$ $1234 / 10 = 123$	$i=2$ $123 \% 10 = 3$ $sum = 4+3=7$ $123 / 10 = 12$	$i=3$ $12 \% 10 = 2$ $sum = 7+2=9$ $12 / 10 = 1$	$i=4$ $1 \% 10 = 1$ $sum = 9+1=10$ $1 / 10 = 0$
---	--	---	--

Logic

$no \% 10 \rightarrow$ to get units place in given number

$no / 10 \rightarrow$ to remove the identified no from given no.

* WAP to find the sum of the squares of the digits given four digit number

{

int no = 1234;

int r = 0;

int sum = 0;

int square = 0;

for (int i=1; i<=4; i++)

{

r = no % 10;

square

square = r * r;

 sum = ~~square~~ sum + square;

no = no / 10;

}

cout < ("Sum of squares of digits " + sum);

}

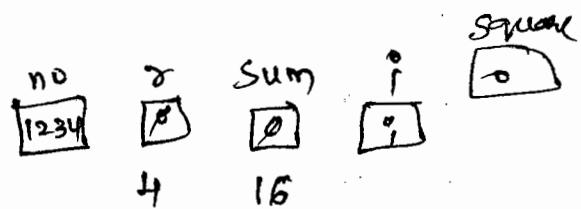
* WAP to find the sum of the cubes of the digits of the given 4 digit no */

* WAP to check whether the given no is Armstrong no or not. (* WAP to reverse the given no*)

Armstrong no:- Armstrong no means sum of the cubes of the digits must be equal to the given no.

e.g:- $153 \Rightarrow 3^3 + 5^3 + 1^3$ Given no = sum

$$27 + 125 + 1 = 153 //$$



29th June 2015 {

{

int no=1234;

int r=0;

int sum=0;

int cube=0;

for (int i=1 ; i<=4; i++)

{

r=no%10;

cube = r*r*r;

sum= sum+cube;

no= no/10;

}

CWL ("Sum of cubes of given no digits is " + sum);

{

~~temp = no~~

check whether Armstrong (or) not

CWL ("Enter a ^{4 digit} no");

int no= int.parse(CRL());

int temp=no; int sum=0; int cube=0; int r=0;

for (int i=1 ; i<=4 ; i++)

{

r=no%10;

cube = r*r*r;

sum= sum+cube

no= no/10; }

```
if (temp == sum)
    C-WL ("No is Armstrong no");
else
    C-WL ("Not an Armstrong no");
```

/* Reverse the ~~no~~-digit no */

{

int no = 1234;

int r = 0;

int revno = 0;

for (int i = 1; i <= 4; i++)

{

r = no % 10;

revno = revno * 10 + r;

no = no / 10;

}

C-WL (revno);

Palindrome :- If the given no is equal to the reverse of the given no.

WAP to check whether the given no is palindrome or not

{

int no=100;

int r=0;

int revno=0;

int temp=no;

for (int i=1; i<=4; i++)

{

r=no%10;

revno=revno*10+r;

no=no/10;

}

c.wl(revno);

if (temp==revno)

{

c.wl("palindrome")

}

else

{

c.wl("Not palindrome")

}

/* WAP to print fibonacci series */

{

1 1 2 3 5 8 13 21

int f1 = 1;

int f2 = 1;

int f3 = 0;

Console.WriteLine(f1 + " \t");

Console.WriteLine(f2 + " \t");

for (int i = 1; i <= 6; i++)

{

f3 = f1 + f2;

Console.WriteLine(f3 + " \t");

f1 = f2;

f2 = f3;

}

Console.WriteLine();

}

\t → gives tab space

\t

Console.WriteLine

(Print the statement
and show the
cursor in sameline)

/* WAP to print the no 12345 */

for (int i = 1; i <= 5; i++)

Console.WriteLine(i + " \t");

```
for (int i=1; i<=6; i++)  
    C.WL(i);
```

O/P:- 1

2

3

4

5

6

```
for (int i=2; i<=16; i+=2)  
    C.WL(i);
```

O/P:- 2

4

6

8

10

12

14

16

```
for (int i=1; i<=19; i+=2)  
    C.WL(i);
```

O/P:- 1
3
5
7
9
11
13
15
17
19

```
for (int i=-5; i<=5; i++)  
    C.WL(i);
```

O/P -5
-4
-3
-2
-1
0
1
2
3
4
5

```
for (int i=10; i>=1; i++)  
    C.WL(i);
```

O/P:- 10
9
8
7
6
5
4
3
2
1

```
for (int i=12; i>=0; i--)  
    C.WL(i);
```

* To print 12345 */

```
for (int i=1; i<=5; i++)  
    C·w(i);
```

* To print 12345 for 5 times */

```
for (int j=1; j<=5; j++)
```

{

```
    for (int i=1; i<=5; i++)
```

{

```
    C·w(i);
```

}

```
    C·wL();
```

}

O/P 12345

12345

12345

12345

12345

* To print 54321 for 5 times */

{

```
for (int j=1; j<=5; j++)
```

{

```
    for (int i=5; i>=1; i--)
```

{

```
    C·w(i);
```

}

{

```
    C·wL();
```

}

O/P 54321

54321

54321

54321

54321

```

for (int i=1 ; i<=5 ; i++)
{
    for (int j=1 ; j<=i ; j++)
        cout << j
    cout << endl;
}

```

Output:

```

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

```

```

for (int i=1 ; i<=5 ; i++)
{
    for (int j=1 ; j<=i ; j++)
        cout << "*"
    cout << endl;
}

```

When $i=1$ rotate the
(j) inner loop one time
when $i=2$ rotate 2 times
 $i=3$ " 3 times
 $i=4$ " 4 times
 $i=5$ " 5 times

```
for (int i=5; i>1; i--)  
{  
    for (int j=1; j<=i; j++)  
    {  
        cout << "*" + "t";  
    }  
    cout << endl;  
}
```

(* factorial no *)

30th June 2015.

```
cout << "Enter a no";  
int no = int::parse (cin);  
int f = 1;  
for (int i=1; i<=no; i++)  
{  
    f *= f * i;  
}  
cout << f;
```

* WAP to Sum of first and last digits in a no/

{

int temp = 0; int sum=0;

int r=0;

r=n0%10;

for (int i=1; i<=4; i++)

{

temp=n0;

n0=n0/10;

}

Sum=r+~~temp~~;

CWL ("sum is" + sum);

While-loop Syntax :-

Initialization;

While (condition)

{

Statements;

increment/decrement;

}

/* WAP to print 1 to 10 no's using while loop.

{

int i=1;

while (i<=10);

{

C.WL(i);

i++;

}

/* WAP to print even no's from 2 to 12 using while loop

{

int i=2;

while (i<=12);

{

C.WL(i);

i=i+2;

}

/* WAP to print no's 10 to 1

{

int i=10;

while (i>=1);

{

C.WL(i);

i--;

}

WAP to reverse any number *!. Using while loop.

```
using System;
class Revdemo
{
    static void Main()
    {
        Console.WriteLine("Enter a no");
        int no = int.Parse(Console.ReadLine());
        int r = 0; int revno = 0;
        while (no != 0)
        {
            r = no % 10;
            revno = (revno * 10) + r;
            no = no / 10;
        }
        C.WL ("Reverse no is " + revno);
    }
}
```

do-while Loop Syntax :-

initialization ;

do

{

Statements ;

inc/dec ;

}

while (Condition) ;

What is diff b/w while and do while loop ?

* while loop will first check the condition and execute the statements.

* do while loop will first execute the statements and then check condition.

* WAP to print only vowels using do-while loop */

Using System ;

class A

{

 static void Main()

 { do {

 C.WL ("Enter a character");

 char ch= char.parse (C.RL());

 switch () {

{

Case 'a':

C-WL ("vowel");

int flag = 0;

do

{

switch (ch) C-WL ("Enter a character");

char ch = Char.Parse (C-RL());

Switch (ch)

Case 'a':

flag = 0;
C-WL ("vowel");

break;

'e'

'i'

'o'

Case 'u':

flag = 0;
C-WL ("vowel");

break;

default :

flag = 1;

C-WL ("Enter a ^{vowel} character")

break;

}

while (flag == 1)

OOPS

1st July 2015

OOPS is a concept which is used to write computer programs by using classes and objects.

* Grady Booch has invented OOPS.

* Principles of OOPS :-

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism.

(Q) What is Object Oriented programming language?

Any language that supports all the principles of OOPS then that language is called as Object Oriented programming language. (OOPL)

e.g:- java, c# .net, vb.net.

• in .net represents .net supports OOPS features.

(Q) What is Object based Programming language?

Any language that does not support atleast one principle of OOPS then that language is called as Object based programming language.

(Q) Why we are learning OOPS?

OOPS is learnt for efficient memory management

What is Abstraction?

It is a process of getting the required data and leaving unnecessary data.

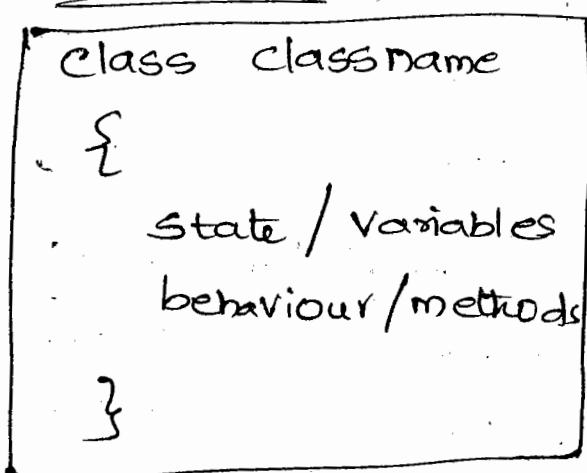
What is Encapsulation?

Encapsulation is a process of binding (or) Wrapping (or) Grouping of state and behaviour in a container.

** In object oriented programming languages like Java, C# .NET we can achieve encapsulation practically by using a keyword called "Class".

** State will have some value and behaviour : is used to perform some operation.

Syntax for class :-



e.g:- Class Student
{
 int sno;
 String sname;
 int m1; int m2; int m3;
 void calculate Totalmarks();
 void Display Total Marks();
}

What is class ?

2nd July 2015

Class is a userdefined datatype (reference type datatype) which consists of variables and methods.

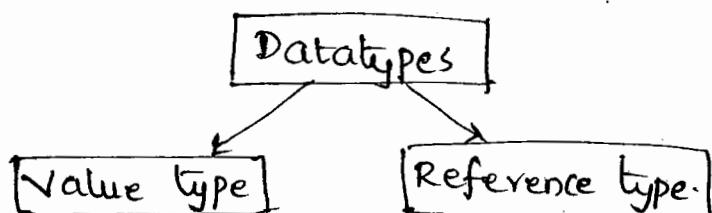
Variables :- Variable is the name given for a particular memory location where the data is located.

What is the purpose of variable ?

The purpose of the variable is to identify the data or to access the data.

What is datatype ?

Datatype specifies the type of data that we store in the memory.



byte-1

short-2

int-4

long-8

float-4

double-8

decimal-16

char-2

string.

Syntax to declare the variable :

datatype variablename;

int end;

end

4 bytes

Variables are of 4 types:-

1. instance variables
2. static variables
3. method parameters
4. local variables.

Instance Variables :- These variables can be declared inside the class and outside the method without static keyword

eg:- class student
{
 int Sno;
 String Sname;
 void SetValue()
 {
 }
}

Static variables :- These variables can be declared inside the class and outside the method with static keyword

eg:- class student
{
 static String Ename;
 void SetValue()
 {
 }
}

Method Parameters :- The variables that was declared within the method parenthesis are called method parameters.

e.g:- class Student

{

void calculateTotalMarks (int m1, int m2, int m3)

{

int total = m1 + m2 + m3;

}

local variables

}

method parameters

Local Variables :- The variables that was declared within the Block are called local variables.

e.g:-

{

int total = m1 + m2 + m3;

}

local variables

3rd July 2015

What is Object?

Object is instance of class.

⇒ Instance means allocating sufficient memory space for the instance variables.

When the memory is allocated for instance Variable? The memory for the instance variable will be allocated at the time of creating Object.

what is the diff b/w class and object?

class can exist without object but object cannot exist without class.

⇒ class does not exist physically but "object" will exist physically.

⇒ class does not have any memory but object will have memory.

⇒ object will be created at the time of execution of the program.

When object is created?

A) Compilation time B) Runtime.

Runtime (object is created at runtime).

4th July 2015

Instance Variable

1. Instance Variable must be declared within the class and outside the method.
2. The memory for the instance variable will be allocated at the time of creating object.
3. The memory for the instance variable will be allocated on heap.
4. The memory for the instance variable will be allocated every time when we create object.

Static Variable

1. Static Variable must be declared inside the class and outside the method with static keyword.
2. The memory for the static variable will be allocated at the time of loading the class.
3. The memory for the static variable will be allocated on stack.
4. The memory for the static variable will be allocated only once at the time of loading the class.

5. If the value is different for different objects then make the variable as instance.
5. If the value is common for all the objects then make the variable as static.
6. The scope of instance variable is within the class i.e. we can access instance variable anywhere within the class.
6. The scope of static variable is within the class (throughout).
7. The lifespan of instance variable is until the object is live (or) until the object is destroyed.
7. The lifespan of static variable is until the class is live.
8. We can access the instance variable by using objectname.
8. We can access the static variable by using classname.

What is Object?

Object is instance of class.

⇒ Instance means allocating sufficient memory for it by RAM for the instance variables.

Syntax for Creating an Object :-

Class new Classname();

(or)

Classname Objectname = new Classname();

⇒ new is a dynamic memory allocation operator which is used to create object i.e. the memory will be allocated for instance variables.

Q) Where object is created?

Heap Memory of RAM.

Q) What is Reference Variable?

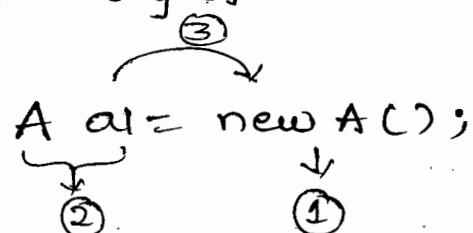
Reference Variable is the name given for a particular memory location where the Object is located.

Q) What is the purpose of reference Variable?

The purpose of reference variable is to access the object data i.e if we want to access the Instance Variable or instance methods, we can access by using Objectname or reference variable.

Q) Where reference variable is created?

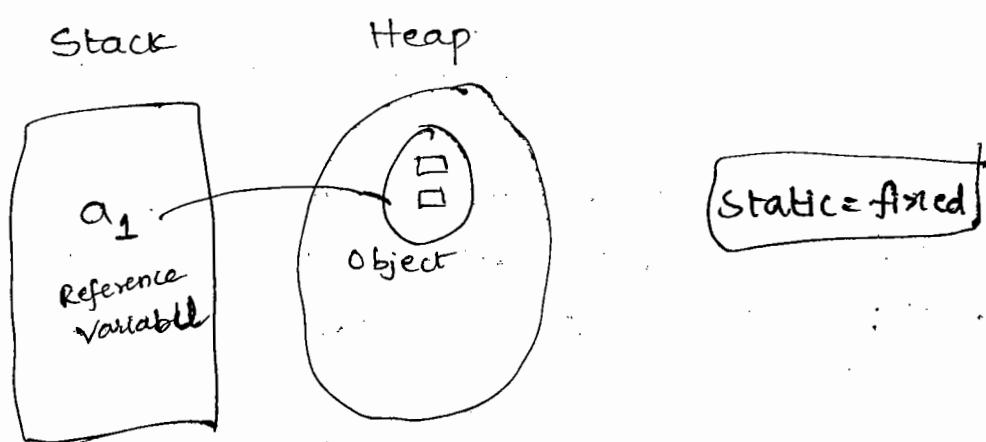
Stack memory of RAM.



Step-1:- Object is created for A class

Step-2:- Reference variable is created for A class

Step-3:- Reference variable is pointing to Object



Classname. Variablename → Syntax to access static variable

• → access operator.

Classname. methodname() → " " static methods

Objectname. Variablename → Syntax to access instance Variable

Objectname. method name() → " " instance methods

Syntax to access static variable ?

Classname. Variablename;

Syntax to access static variable ?
method

Classname. methodname();

Eg:- Console.WriteLine();

↓

Classname

↓

Static method.

Syntax to access Instance Variable ?

Objectname. Variablename;

Syntax to access instance method

Objectname. methodname();

Here . is called as access operator.

What is method ?

Method is a Subprogram which is used to perform some operations.

⇒ Methods are of two types → * Static method

* Instance method.

- ⇒ Static method must be declared with static keyword.
- ⇒ Method is used to write some logic
- ⇒ Method will get executed when we call it.
- ⇒ We can access the static method by using class name.
- ⇒ We can call the instance method by using object name.

Syntax for declaring static method :-

Class classname

```
{  
    static void methodname()  
    {  
        logic  
    }  
}
```

Syntax for declaring instance method :-

Class classname

```
{  
    void methodname()  
    {  
        logic  
    }  
}
```

Sample Program

5th July 2015

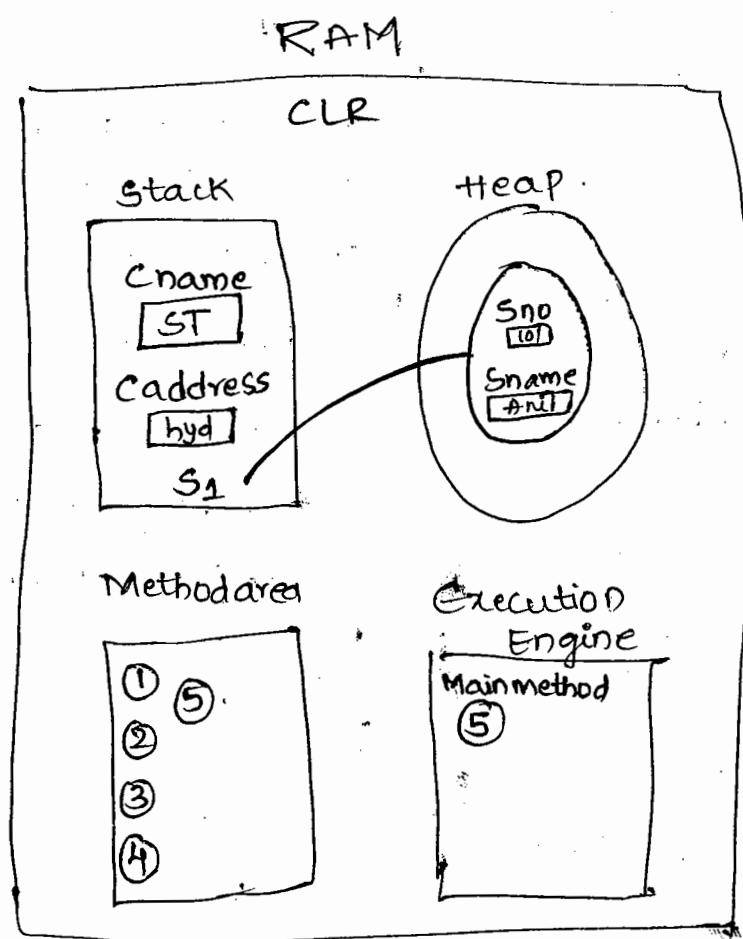
```
using System;
class Student
{
    static string cname;
    static string caddress;
    int sno;
    string sname;
    static void SetCollegeData()
    {
        cname = "Sathya Technologies";
        caddress = "Hyderabad";
    }
}
```

```
void SetStudentData()
{
    sno = 101;
    sname = "Anil";
}
```

```
static void DisplayCollegeData()
{
    C.WL(cname);
    C.WL(caddress);
}
```

```
void DisplayStudentData()
{
    C.WL(sno);
    C.WL(sname);
}
```

```
static void Main()
{
    Student.SetCollegeData();
    Student s1 = new Student();
    s1.SetStudentData();
    Student.DisplayCollegeData();
    s1.DisplayStudentData();
}
```



- * Save the program with Student.cs in E:\sample programs.
- * Compile the program.
CSC Student.cs
- * Whenever we compile the program the compiler will check for syntax errors and compiler will check for Static Void Main()
- * If static void Main() is declared then compiler will display error message.
- * Execute Then Compiler will generate Student.exe.
- * Execute the program.
Whenever we execute the program then the class will be loaded into CLR.

Whenever the class is loaded

Step-4:- CLR will allocate memory for static Variables
on stackmemory.

Step-2 :- Then all the 5 methods including main
method will load into method area.

Step-3 :- CLR will search for static void Main ()
and load into execution engine (EE) and start
executing the program.

Step-4 :- Set CollegeData () [Method 1] is called
from method area and it is executed in EE.

Step-5 :- [Method 2] is called.

Student s1 = new Student ();

* New is dynamic memory allocation operator which will
create object i.e. memory is allocated for instance
variables on heap.

* Student s1 is a reference variable.

The purpose of reference variable is to access the object
data.

* Reference Variable is stored on stack and it is
pointing to the object.

Step-6 :- Method 3 is called and executed.

Step-7 :- Method 4 is called and executed.

Step-8 :- Then Method 5 is removed from the Execution Engine.

Method Parameters :- The variables that was declared within the parenthesis are called as method parameters.

Eg:- class A

```
{  
    void show(int x, int y)  
    {  
        cout << x + y;  
    }  
}
```

```
A a;  
a.show(10, 20);
```

* At the time of calling the method we have to ~~pass~~ the values.

* The no. of values that we ~~pass~~ must match with no. of parameters.

* The order of values that we ~~pass~~ must match with order of parameters.

* The type of values that we ~~pass~~ must match with type of parameters.

⇒ The scope of method parameters is within the method.

⇒ Method parameters are used to ~~pass~~ the values at the time of calling the method.

Heap is unnamed memory so ^{all} reference is stored on Stack

```
class Student
```

```
{
```

```
    static string cname = "satbyatech";  
    static string address = "hyd";  
    int sno;  
    string sname;  
    void SetStudentData(int x, strings)
```

```
{
```

```
    sno = x;
```

```
    sname = s;
```

```
}
```

```
    static void DisplayCollegeData()
```

```
{
```

```
        C.WL(cname);
```

```
        C.WL(address);
```

```
}
```

```
    void DisplayStudentData()
```

```
{
```

```
        C.WL(sno);
```

```
        C.WL(sname);
```

```
}
```

```
    static void Main()
```

```
{
```

```
        Student S1 = new Student();
```

```
        S1.SetStudentData(101, "anil");
```

```
        Student S2 = new Student();
```

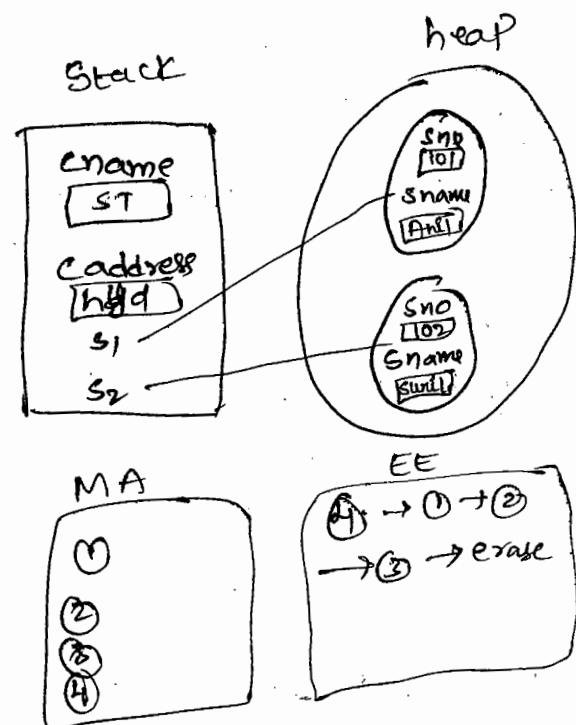
```
        S2.SetStudentData(102, "sunil");
```

```
        S1.DisplayStudentData();
```

```
        S2.DisplayStudentData();
```

```
}
```

```
}
```



Map to declare empno, ename, basic sal, company name, company address.

Calculate D.A, H.RA & total sal.

Class Employee

{

Static String cname = "ST";

Static String cadd = "Hyd";

int eno;

String ename;

double bsal;

double da;

double hra;

double totalsal;

return type

void Set Emp Data (int x, string y, double d)

{

eno=x;

ename=y;

bsal=d;

}

Void calda()

{

da=0.2*bsal;

}

Void calhra()

{

hra=0.4*bsal;

}

Void caltsal()

{

tsal=bsal+da+hra;

}

Void Disp EmpData()

{

c.wl(eno);

c.wl(ename);

c.wl(bsal);

c.wl(da);

c.wl(hra);

c.wl(tsal);

static void Main()

{

Employee e1 = new Employee();

e1.setEmpData(101, "anil", 20000);

e1.calda();

e1.calhra();

e1.caltsal();

e1DispEmpData();

6th July 2015

Constructor:- Constructor is a default method which is used to initialize the default values for instance variables.

* Constructor will get executed ~~at the~~ immediately after creating object.

* Constructor is also a method which does not have return type atleast void.

Rules to declare Constructor:-

1. Constructor name and classname must be same.

2. Constructor does not have return type atleast void.

3. Constructor will get executed immediately after creating object.

Constructor

* Constructor is used to initialize the default values for instance variables.

* Constructor does not have any return type.

* Constructor name and classname must be same.

* Constructor will gets invoked immediately after creating object

* Constructor will be called executed only once at the time of creating object.

Method

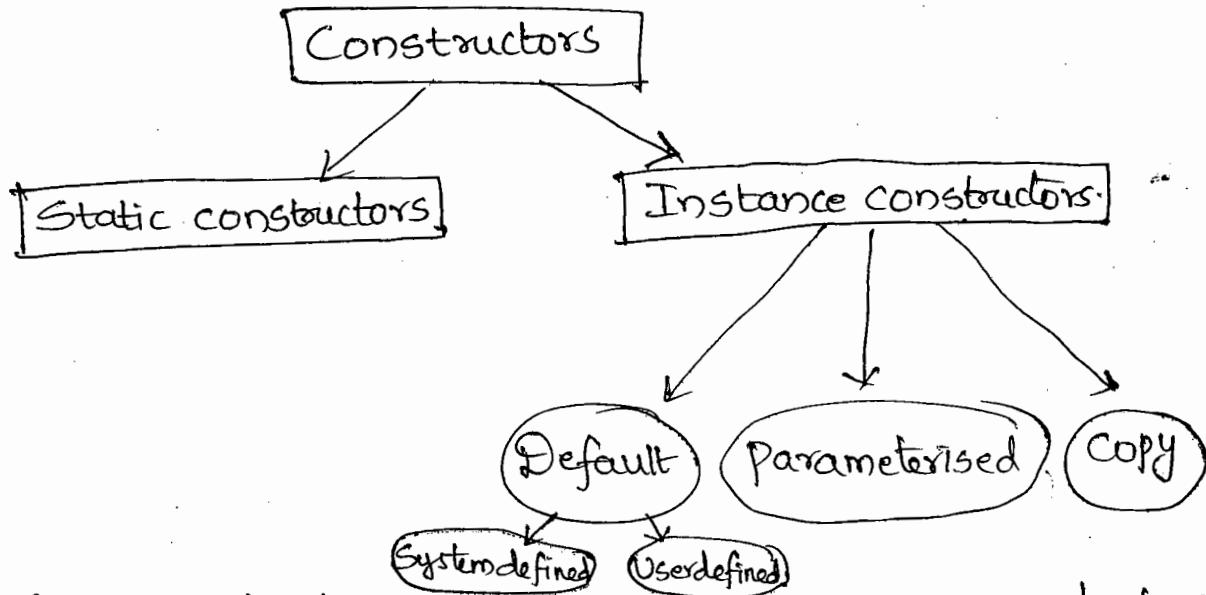
* We cannot initialize the value. It is used to perform some logic or Method is used to assign the values.

* Method must have return type.

* Method name and classname must not be same.

* Method will gets executed when we call it.

* Method will get executed everytime when we call it.



Static Constructor :- Static constructor must be declared with static keyword.

- * static constructor is used to initialize the values for static variables.
- * Whenever the class is loaded , CLR will allocate memory for static Variables on stack.
- * All the methods including constructors will load into method area.
- * CLR will Search for Static constructors and it will execute the code that was written within the static Constructor.
- * Then CLR will Search for Static void Main Method and Start executing the program.
- * the call to the static constructor is done by CLR.

Syntax for declaring Static Constructor :-

```
Class classname  
{  
    Static classname()  
    {  
    }  
}
```

What is the output of the below program?

```
using System;  
Class A  
{  
    static AC()  
    {  
        Console.WriteLine(" I am static constructor");  
    }  
    static void Main()  
    {  
        Console.WriteLine(" I am Main Method");  
    }  
}
```

Output:- I am static constructor

I am Main Method.

7th July 2015

Instance Constructor :-

It is used to initialize the default values for the instance variables.

Instance constructor is of three types.

Default Constructor :- Default constructors is of two types.

* System defined Constructor

* User defined Constructor

System defined default constructor :- At the time of compilation of program the compiler will check any constructor was defined within the class or not.

* If the constructor was not declared by the programmer then the compiler will declare system defined default constructor and it will initialize default values for instance variables.

Sample program :-

Using System;

Class X

{

int a; int b; string c; char d; double e;

Void Display Data()

{

C.WL(a);

C.WL(b);

C.WL(c);

C.WL(d);

C.WL(e);

}

Compiler will write this
Public X() code internally

{
a=0;
b=0;
c=null;
d='';
e=0.0;

```
Static void Main()
```

```
{
```

```
    X x1 = new X();
```

```
    x1.Display Data();
```

```
}
```

```
}
```

Userdefined default constructor:- It is used to initialize user defined values instead of default values

Sample program:-

```
using System;
```

```
Class Student
```

```
{
```

```
    int Sno;
```

```
    String Sname;
```

```
    Student() → ①
```

```
{
```

```
    Sno=101;
```

```
    Sname="Anil";
```

```
}
```

```
    void Displaydata();
```

```
{
```

```
    C.WL(Sno);
```

```
    C.WL(Anil);
```

```
}
```

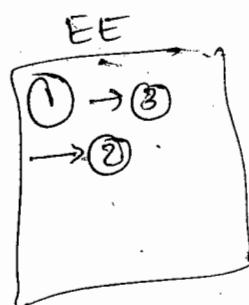
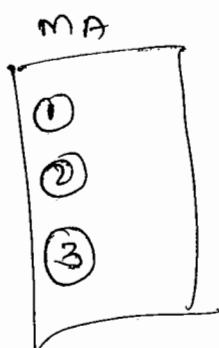
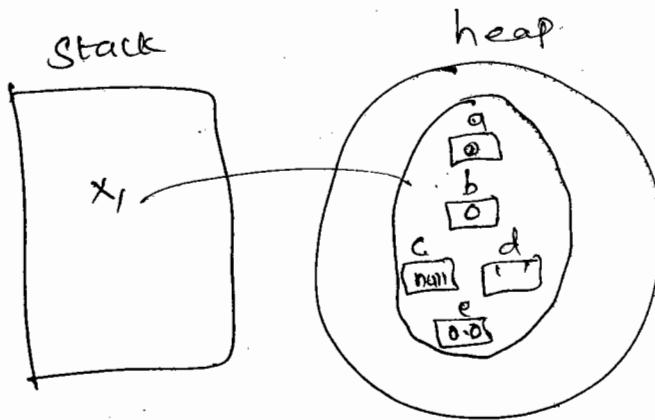
```
Static void Main()
```

```
{
```

```
    Student S1 = new Student();
```

```
    S1.Displaydata();
```

```
} }
```



```
→ ③
```

Note :- In the above example when we create multiple Objects then same values will be initialized for all the objects.

Parameterised Constructor :-

* If we want to initialize different values for different Objects then we have to use parameterised constructor.

* At the time of declaring the constructor we have to declare the parameters and at the time of creating the objects we have to pass the values.

Syntax :- class Classname

```
{  
    Classname (parameters)  
    {  
    }  
}
```

* There is no static parameterised constructor bcoz class cannot give the values.

Sample program:-

Class Student

{

Static String name;

Static String address; // static Variables

int sno;

String sname; // Instance Variables

String address;

static student()

{}

Static Student() ①

{

Cname = "ST"; // static constructor

Cadd = "hyd";

}

Student (int x, String y, String z) ②

{

Sno = x;

// parameterised constructor

Sname = y;

Add = z;

Static void Disp College Data() ③

{

C.WL(Cname + "/" + Cadd);

}

Void Display Student Data() ④

{

C.WL(Sno);

C.WL(Sname);

C.WL(Add);

}

Static void Main() ⑤.

{

Student S₁ = new Student(101, "Anil", "hyd");

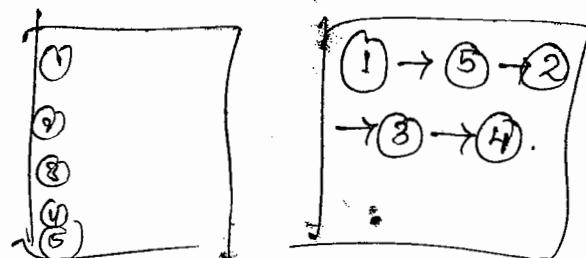
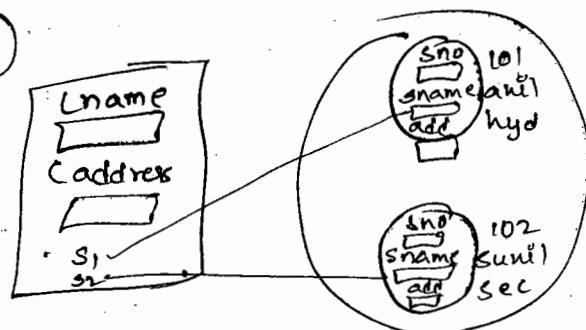
Student S₂ = new Student(102, "sunil", "sec");

S₁.Display Student Data();

S₂.Display Student Data();

}

}



Whenever we execute the program

1. CLR will allocate memory for static Variables on Stack.
2. All the methods including static constructor and parameterised constructor will load into Method area (3 methods + 2 cons)
3. CLR will search for static constructor and load in EE and static constructor will initialize the values for static variables.
4. CLR will search for static void Main() and start executing the program.
5. Student s1 = new Student(101, "Anil", "Hyd");
 - * new is a dynamic memory allocation operator which will create object i.e. the memory will be allocated for instance variables on heap. Student(101, "Anil", "Hyd")
 - * Parameterised constructor is invoked and it will initialise the userdefined values for instance variables.
 - * Reference variable is created on stack and it will point to object

8th July 2015

What is constructor overloading?

It is process of writing more than one constructor with same name but with different parameters.

e.g. Class A

```
{  
A()  
{  
CWL("I am D.C.");  
}  
A(int x)  
{  
CWL("I am SPC");  
}
```

```
A (int x, int y)  
{  
CWL("I am dpc");  
}  
static void Main()  
{  
new A();  
new A(10);  
new A(10,20); } }
```

oops casestudies

1. Class Bankaccount

```
{  
    static string bname;  
    static string brname;  
    string dname;  
    long accno;  
    string acctype;  
    double cbal;  
  
    static Bankaccount()  
    {  
        bname = "Axis";  
        bname = "hyd";  
    }  
}
```

```
Bankaccount(string dname,  
            long accno, string acctype, double cbal)  
{  
    this.dname = dname;  
    this.accno = accno;  
    this.acctype = acctype;  
    this.cbal = cbal;  
}
```

void deposit(double amt)

```
{  
    cbal = cbal + amt;  
}
```

void withdraw(double amt)

```
{  
    if (amt <= cbal)  
    {  
        cbal = cbal - amt;  
    }  
}
```

static void displayBankdata()

```
{  
    c.wl(bname + "/" + brname);  
}
```

void displayCustdata()

```
{  
    c.wl(cname);  
    c.wl(accno);  
    c.wl(cbal);  
}
```

static void Main()

```
{  
    Bankaccount b1 = new Bankaccount(  
        ("anil", 1000000001, "savings", 5000));  
}
```

BA b2 = new Bankaccount

```
(("sunil", 100000002, "savings", 3000));
```

b1. deposit(2000);

b1. withdraw(1000);

b1. displayCustdata();

b2. deposit(2000);

b2. withdraw(2000);

b2. displayCustdata();

}

}

Eg for constructor overloading :-

```
using System;
```

```
class Person
```

```
{
```

```
    String fname;
```

```
    String lname;
```

```
    String emailid;
```

```
    DateTime dob;
```

```
Person (String fname, String lname, String @mailid)
```

```
{
```

```
    this.fname = fname;
```

```
    this.lname = lname;
```

```
    this.emailid = emailid;
```

```
}
```

```
Person (String fname, String lname, DateTime dob)
```

```
{
```

```
    this.fname = fname;
```

```
    this.lname = lname;
```

```
    this.dob = dob;
```

```
}
```

```
3
```

What is this ?

this is a keyword which is declared when those
method parameters are equal to instance variables names.

this is used to access the instance variables

in the constructor before creating reference variable

i.e when the constructor is called.

What is this ?

9th July 2015

This is a keyword which represents the Current Class Object.

When to use this ?

If we want to access the Instance Variables after Creating object and before getting reference then we have to use this.

Method:-

Method is used to perform some operation.

Method = Method heading + Method body.

Method heading = Access modifier + method name + parameters.

Method body = The code that we write within the block

is called as method body.

Access modifier return type method name Parameters
 ↑ ↑ ↑

Public void Add (int x, int y)

→ Method heading.

{

→ Method area

}

What are method parameters ?

The variables that we declare within the parenthesis are called as method parameters.

(int x, int y).

Method heading → what to do ?

Method body → how to do ? we write the code in body.

Q) What is method signature?

Method Signature = Method name + Parameters.

* Method must have return type.

* The return type may be void or datatype.

Syntax :-

return type methodname (parameters)
 ↑
 datatype
 {
 }
 }

* If the method is not returning any value then the return type must be void.

* If the method is returning the value then the return type is datatype.

eg:- Class A

```
{  
void show()  
{  
    CWL("iam show"); } }
```

A a1=new A();
a1.show();

Void show()

```
{   ✓ Valid  
}
```

Void show()

```
{  
    return 10; not valid.  
}
```

eg:- Class A

{ return type
 int Add(int x,int y)

```
{  
    return x+y;  
}  
    }   return value
```

A a1=new A();

```
int result=a1.Add(10,20);  
CWL(result);
```

void show() { } Valid.

void show() { }
return 10;
(not valid)

int show() { }
(not valid)
return value is required.

int show() { }
return 10; Valid.
}

int show() { }
return "10";
(not valid)

String show() { }
return "sathya";
Valid.

String show(int x) { }
(valid)
return "10"+x;

Class A

{
String display()
{
return "Welcome Sathya";
}}

A a1 = new A();
String s = a1.display();
CWL(s); (or)

CWL(a1.display());

```
Class A
{
    A show()
    {
        return new A();
    }
}

A a1 = new A();
A a2 = a1.show();
```

```
Class A
{
    void show()
    {
        C.WL("I am show");
    }

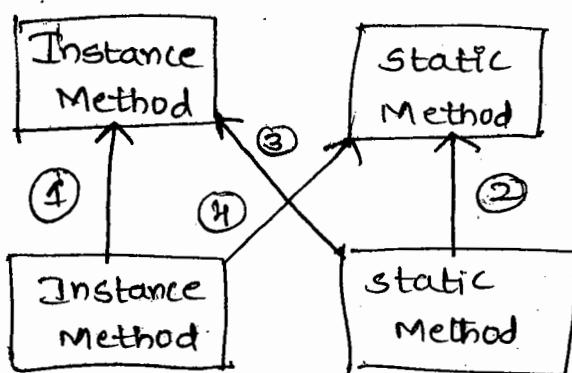
    A display()
    {
        return new A();
    }

    A a1 = new A();
    A a2 = a1.display();
    a2.show();
```

```
Class Sql Command
{
    int ExecuteNonQuery() { return 0; }
    object ExecuteScalar() { return 10; }
    SqlDataReader ExecuteReader() { }
}

Sql Command Obj = new Sql Command();
int i = Obj.ExecuteNonQuery();
object o = Obj.ExecuteScalar();
SqlDataReader dr = Obj.ExecuteReader();
```

10th July 2015



1. Can we call one instance method in another instance method in the same class?

Whenever we call an IM in another IM objectname is not required.

* We can directly access the IM in another IM without using objectname.

Eg:- class A

{

Void Show()

{

C.WL ("I am show");

}

Void Display()

{

Show();

C.WL ("I am display");

}

Static Void Main()

{

A a1 = new A();

a1.Display();

}

}

2. Can we call a SM in another SM directly?

Yes we can directly access the SM in another SM without using classname within the same class using ~~class~~ System;

```
Class A
{
    Static Void Show()
    {
        C WL ("I am show");
    }
    Static Void Display()
    {
        Show();
        C WL (" I am display");
    }
    static void Main()
    {
        Display();
    }
}
```

3 Can we call IM in SM directly?

We cannot directly call IM in SM.

* If we want to access Instance Method in SM Objectname is required.

Eg:- We are calling every IM in Main Method which is of Static Method by using Objectname.

Eg:- Using System;

```
Class A
{
    void Show()
    {
        C WL ("I am show");
    }
}
```

```
Static Void Display()
{
    A al = new A();
    al . Show();
}
```

```
Static Void Main( )
```

```
{
```

```
    Display( );
```

```
}
```

```
}
```

Q. Can we call SM in IM directly ?

Yes we can directly access SM in IM within the same class.

Eg:- using System;

Class A

```
{
```

```
Static Void Show( )
```

```
{
```

```
C.WL ("I am show");
```

```
}
```

```
Void Display( )
```

```
{
```

```
C.WL ("I am display");
```

```
Show();
```

```
-
```

```
Static Void Main( )
```

```
{
```

```
C.WL ("I am Main");
```

```
A a1 = new A();
```

```
a1.Display()
```

```
}
```

```
}
```

Can we call a method inside a constructor?

Yes, we can call a method inside a constructor.

* The method will get executed when the constructor is invoked.

* Whenever we call a method inside a constructor that method must consist of Initialization logic because constructor contains Initialization logic only.

Eg:- Using System;

Class A :

```
{  
    A()  
    {  
        System.out.println("I am constructor");  
        show();  
    }  
    void show()  
    {  
        System.out.println("I am show");  
    }  
    static void Main()  
    {  
        new A();  
    }  
}
```

Inheritance :-

11th July 2015

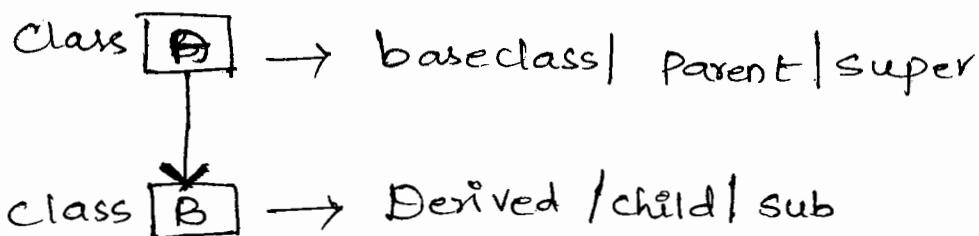
Inheritance is used to establish the relationship between two or more classes.

* It is a mechanism of creating a new class by already existing class.

* It is a mechanism of obtaining variables and methods from one class to another class.

* The class which is giving variables and methods is called as Base class or Super class or Parent class.

* The class which is taking variables and methods is called as Derived class or Subclass or Child class.



* The operator : is called as Inheritance Operator.

: means extends.

Class A

{
2V
3M
3C
}

No. of Variables in A :- 2

No. of Methods in A :- 3
No. of Constructors in A :- 3

Class B : A

{
3V
2M
2C
}

No. of Variables in B :- 5
No. of Methods in B :- 5
No. of Constructors in B :- 2

Does Constructors will Participate in Inheritance?

No, they does not participate in Inheritance.

What are the advantages of Inheritance?

* Reusability

* Extensibility

* Reimplementation.

Note:- In Inheritance always create object for derived class.

Points to remember on Inheritance :-

Point-1:- Whenever we create an object for derived class then memory will be allocated for both derived class and base class Instance Variables.

Point-2:- The derived class object can referred either by using base class name or derived classname.

Point-3:- When the derived class object is referred by using base class name we can access only base class variables and methods.

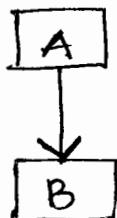
* When the derived class object is referred by using derived class name we can access both base class and derived class variables and methods.

Types of Inheritance:-

12th July 2015

- * Single level Inheritance.
- * Multi level Inheritance.
- * Hierarchical Inheritance.
- * Hybrid Inheritance.
- * Multiple Inheritance.

Single level Inheritance :- It is a process of creating a Derived class by using base class.



Sample program:-

```
using System;
class calculator2000
{
```

```
    public void Add(int x, int y)
```

```
{
```

```
    CWL("sum is" + (x+y));
```

```
}
```

```
    public void Sub(int x, int y)
```

```
{
```

```
    CWL("Diff is" + (x-y));
```

```
}
```

```
    public void Mul(int x, int y)
```

```
{
```

```
    CWL("Product is" + (x*y));
```

```
}
```

```
public void Div (int x, int y)
{
    C.WL ("Quotient is" + (x/y));
    C.WL ("Remainder is" + (x%y));
}
```

```
class Calculator2015 : Calculator2000
{
```

```
    public void square (int x)
    {
        C.WL ("Square is" + (x*x));
    }
```

```
    public void cube (int x)
    {
        C.WL ("Cube is" + (x*x*x));
    }
```

```
    public void sqrt (int x)
    {
        C.WL ("Sqrt is" + (Math.sqrt(x)));
    }
```

```
    public void power (doublebase no, doubleexponent)
    {
        C.WL ("Power is" + (Math.Pow(base no, exponent)));
    }
}
```

```
class program
```

```
{  
    static void Main()
{
```

```
calculator2015 obj = new Calculator2015;  
obj.Add (30,10);  
obj.Sub (30,10);
```

```
Obj. MUL(2,3);  
Obj. DIV(5,3);  
Obj. Square(5);  
Obj. Cube(5);  
Obj. Sqrtn(25);  
Obj. Power(3,5);  
}  
}
```

Multi-level Inheritance :- It is a process of creating
a derived class by using another derived class.



Sample program :-

```
using System;  
class A  
{  
    public void Show()  
    {  
        Console.WriteLine("I am Show");  
    }  
}  
class B : A  
{  
    public void Display()  
    {  
        Console.WriteLine("I am Display");  
    }  
}
```

```

class C:B
{
    public void Print()
    {
        C.WL ("I am print");
    }
}

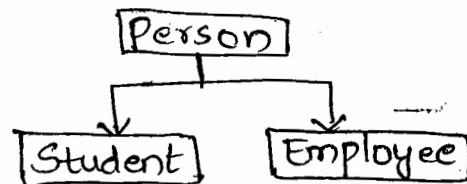
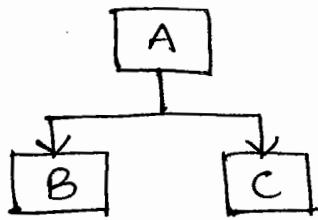
```

```

class Program
{
    static void Main()
    {
        C c1 = new C();
        c1.Show();
        c1.Display();
        c1.Print();
    }
}

```

Hierarchical Inheritance:- It is a process of creating more than derived class from a single base class.



Student is a person
Employee is a person

Sample program:-

Class Program

using System;

Class A

```

{
    public void Show()
    {

```

```

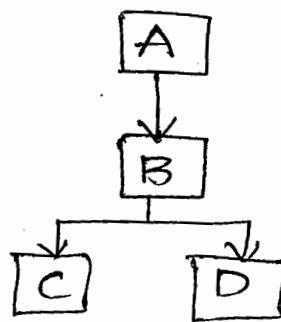
        C.WL ("I am show");
    }
}
```

```
Class B:A  
{  
    Public void Display()  
    {  
        C.WL ("I am display");  
    }  
}
```

```
Class C:A  
{  
    Public void print()  
    {  
        C.WL ("I am print");  
    }  
}
```

```
Class program  
{  
    Static Void Main()  
    {  
        B b1 = new B();  
        b1.Show();  
        b1.Display();  
  
        C c1 = new C();  
        c1.Show();  
        c1.print();  
  
    }  
}
```

Hybrid Inheritance* :- It is the combination of any two inheritance.



Constructors participation with Inheritance:-

Q) Does constructor will participate in inheritance? -

No, Constructors will not participate in inheritance.

What is this() ?

this() is used to call the current class Default Constructor.

What is this(~~x,y~~) → this(10) ?

this(10) is used to call the current class single Parameterised constructor.

What is this(~~x,y~~) → this(10,20)?

this(10,20) is used to call the current class double Parameterised constructor.

Note :- Always the Constructors will be called from Bottom to Top and execute from top to bottom.

* If the above rule is not followed then Recursion Problem occurs (i.e there will be no end for execution).

Sample program:- How to call the current class constructor
by using this()?

using System;

Class A

{

public A()

{

c.wl ("I am DC");

}

public A(int x): this()

{

c.wl ("I am SP");

}

public A (int x, int y): this(10)

{

c.wl ("I am DPC");

}

}

Class program

{

Static void Main()

{

New A(10,20);

}

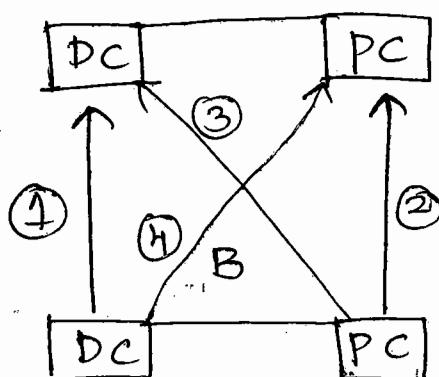
}

base() :- It is used to access the base class default constructor.

base(10) :- It is used to access the base class SP.C.

base(10,20) :- It is used to access the base class DP.C.

A



Q. How to invoke the default constructor of base class by crea w.r.t default constructor of derived class?

Using System;

Class A

{

public A()

{

c.wL (" I am A class DC");

}

}

Class B: A

{

public B(): base()

{

c.wL (" i am B class DC");

}

}

class program

{

static void Main()

{

new B();

}

}

base() is
Optional bcoz
Compiler will write

* If the programmer does not write base() in derived class then compiler will write base() while compilation.

Q) How to invoke parameterised constructor of base class w.r.t to parameterised constructor of derived class?

Using System;

Class A

```
{  
public A(int x)  
{  
cWL("i am A class SPC");  
}  
}
```

Class B:A

```
{  
public B(int y):base(10)  
{  
cWL(" I am B class SPC");  
}  
}
```

Class program

```
{  
Static void Main()  
{  
new B(10);  
}  
}
```

base(10) is mandatory bcoz compiler will write only base()

Q) How to invoke the default constructor of base class w.r.t parameterised constructor of derived class.

Using system;

Class A

```
{  
public A ()
```

```
{  
    C::WL (" I am DC");  
}  
}
```

Class B:A

Here base() is optional.

```
{  
public B (int x): base()  
{  
    C::WL (" I am SPC");  
}  
}
```

Class program

```
{  
Static void Main()  
{  
    new B (10);  
}  
}
```

Q) How to invoke pc of base class w.r.t DC of derived class?

Using system;

Class A

```
{
```

```
Public int(x)
{
    C-WL (" I am spc");
}
}

class B: A
{
public B() : base(10)
{
    C-WL (" I am B class DC");
}
}
```

Here base(10) is mandatory bcoz compiler will write only base().

Class program

```
{}
Static void Main()
{
    new B();
}
```

Class A

{

public A() {

{

c.wl("I am A DC");

}

public A(int x) : this()

{

c.wl("I am A SpC");

}

public C(int x, int y) : this(10)

{

c.wl("I am A DPC");

}

Class B: A

public B() : base(10, 20)

{

c.wl("I am B DC");

}

public B(int x) : ~~base~~(~~x~~) this()

{

c.wl("I am B SpC");

}

public B(int x, int y) : ~~base~~(~~x, y~~) this(10)

{

c.wl("I am B DPC");

}

Class program

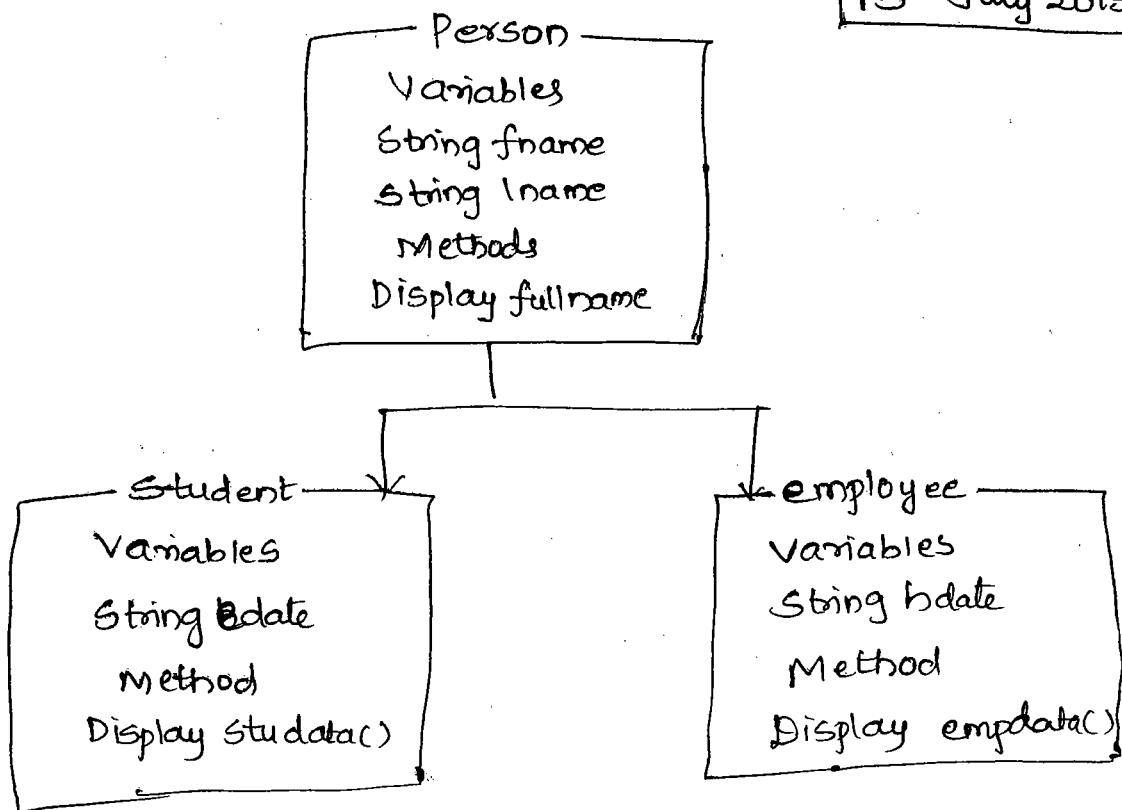
{

static void Main()

{

new B(10, 20);

13th July 2015



Using system;

Class Person

{
 String fname, String lname;

Public person (String fname, String lname)

{

 this.fname=fname;

 this.lname=lname;

}

Public void Display fullname()

{

 c.wr (fname + lname); } }

Class Student: Person

{

 String edate;

}

```
Public Student( string fname, string lname, string edate)
    : base (fname, lname)
```

```
{  
    this.edate = edate; }
```

```
Public void Display studentData()
```

```
{  
    base.Display FullName();  
    CWL("edate is " + edate); }
```

```
Class employee: Person
```

```
{  
    string hdate;
```

```
Public employee( string fname, string lname, string hdate ) :
    base (fname, lname)
```

```
{  
    this.hdate = hdate; }
```

```
Public void Display employeeData()
```

```
{  
    base.Display fullname();
```

```
CWL("hiredate is " + hdate); }
```

```
Class program
```

```
{ static void Main()
```

```
{  
    Student s1 = new Student("Anil", "Kumar", 2-3-15);
```

```
    s1.Display studentData();
```

```
    employee e1 = new employee ("sunil", "kumar", 3-5-12);
```

```
    e1.Display employeeData();
```

```
}
```

```
}
```

Sealed class :- Sealed class will not participate in Inheritance.

Q) How to stop Inheritance ?

By making the class as sealed class.

Syntax :- Sealed class classname
{

}

* Sealed class consists of variables, methods, constructors etc.

Eg:- using System;

Sealed Class A

{

public void Show()

{

CWL ("I am Show");

}

}

Class program.

{

Static Void Main()

{

A a1 = new A();

a1.Show();

}

}

Class A
{
}
(valid)

Sealed Class A
{
}
X
Class B : A
{
}
(Invalid)

Sealed class B : A
{
} ✓
}

Partial class :- Partial class must be declared with
partial keyword. It participate in Inheritance.

* It is a process of writing more than one class
with same name.

* At the time of compilation all the partial classes
having same name will combine as a single class.
* The main advantage of introducing partial classes is
in ASP.NET the designing code is available in one
partial class and the business logic code is available
in another partial class. At the time of compilation both
the classes will become as single class.

Eg:- Partial class A

```
{  
    public void show()  
    {  
        C.WL("I am show");  
    }  
}
```

At compilation ESE A.cs

```
class A  
{  
    public void show() {}  
    public void display() {}  
}
```

Partial class A

```
{  
    public void Display()  
    {  
        C.WL(" I am Display");  
    }  
}
```

Execution:-
class program

```
{  
    static void Main()  
    {  
        A a1 = new A();  
        a1.show();  
        a1.Display();  
    }  
}
```

Static class :- Static class must be declared with static keyword.

* Static class consists of only numbers like static variables, static methods, static constructors.

Q) When to declare a class as static?

If the class consists of only static numbers then make that class as static class.

Eg:- using System;

static class date

```
{  
    static int day;  
    static int month;  
    static int year; }
```

static Variables

Static Date() → static constructor.

```
{
```

day = System.DateTime.Now.Day;

month = System.DateTime.Now.Month;

year = System.DateTime.Now.Year;

```
}
```

static void Display Date() → static method

```
{
```

Console.WriteLine("{0}-{1}-{2}", day, month, year);

```
}
```

static void Main() → static main method

```
{
```

Display Date();

```
}
```

```
}
```

Q) Can we Create an Object for static class?

No.

Q) Does static class participate in Inheritance?

No.

14th July 2015

Polymorphism:-

Greek poly + morphos \Rightarrow many forms
words

* Polymorphism came from two greek words

poly + morphos
↑ many forms

* Polymorphism is defined as anything that exhibit
in multiple forms is called as polymorphism.

e.g:- + operators will perform Addition operation between
2 integers and Concatenation operation between
2 strings.

* In Object Oriented programming Languages like Java or
~~C++~~ we can achieve Polymorphism by using
Overloading and Overriding.

* Overloading is of 3 types:-

1. Method overloading
2. Constructor overloading
3. Operator overloading

Method Overloading :- It is a process of writing more than one method with same method name but with different parameters.

Method overloading = Same method name + different parameters

* In overloading the no. of parameters, order of parameters and type of parameters must differ.

* In overloading the method call will be done based on the input values that we pass.

Eg:- using System;

Class A

{

public void Add (int x, int y)

{

C.W.L ("Sum of 2 no's is " + (x+y));

}

public void Add (int x, int y, int z)

{

C.W.L ("Sum of 3 no's is " + (x+y+z));

}

Public void Add (String x, String y)

{

C.W.L ("String sum " + (x+y));

}

Class program

{

Static void Main ()

{

A a1 = new A();

a1.Add(10, 20);

a1.Add("sathy", "Tech");

a1.Add(10, 20, 30);

}

}

{

Inheritance based overloading :-

Using System;

Class A

{

Public Void show()

{

CWL (" I am without parameters");

}

Public void show (int x)

{

CWL (" I am with single Parameter");

}

Class B : A

{

Public Void show (int x, int y)

{

CWL (" I am with 2 Parameters");

}

Class program

{

Static Void Main()

{

B b1 = new B();

b1.show();

b1.show(10);

b1.show(10/20);

}

}

Can we achieve overloading by using methods?

Yes, static methods will participate in overloading.

* Best example for overloading is C.WL();

Class console

{

public static void WriteLine() { }

" " (bool value) { }

" " (int value) { }

" " (string value) { }

" " (double value) { }

" " (float value) { }

----- 19 methods

}

What is constructor overloading?

It is a process of writing more than one constructor with same name but with different parameters.

OVERRIDING :-

15th July 2015

It is a process of reimplementing the base class method in the derived class.

Q) When to go for overriding?

If the base class method is not satisfying the requirement of derived class then we have to override the base class method in the derived class.

Rules for overriding :-

1. Inheritance is mandatory
2. Same method heading + different method body in base class and derived class.
3. Virtual in base class and override in derived class
4. In overriding always the overridden method i.e derived class method gets executed.

e.g:- class A

{

 public virtual void show() {}

}

class B : A

{

 public override void show() {}

}

Does overriding is possible with single class?

No, Inheritance must be there.

Does overriding is possible with static methods?

No, overriding is not possible with static methods.

Is it mandatory to override virtual methods?

Virtual methods may or may not override.



* Overriding ~~is~~ virtual methods is optional.

* Overriding abstract methods is mandatory.

Eg:- using system;

Class A

```
{  
    public virtual void show()  
    {  
        C.WL("i am A show");  
    }  
}
```

Class B:A

```
{  
    public override void show()  
    {  
        C.WL("i am B class show()");  
    }  
}
```

class program

```
{  
    static void Main()  
    {  
        B b1 = new B();  
        b1.show();  
    }  
}
```

Output:- I am B class show

e.g:- using System;

Class RBI .

{

public virtual void show() .

{

c.wl ("ROI is 3%");

}

public void Deposit (double amount)

{

c.wl ("AMT Deposited");

}

public void withdraw (double amount)

{

c.wl ("AMT withdrawn");

} }

Class Axis : RBI

{

public override void show()

{

c.wl ("ROI is 5%");

} }

Class ICICI : RBI

{

public override void show()

{

c.wl ("ROI is 7%");

} }

Class SBI : RBI

{ }

Class program

{

static void Main()

{

Axis a1 = new Axis();

a1. Show();

a1. Deposit();

a1. withdraw();

ICICI a2 = new ICICI();

a2. Show();

a2. Deposit();

a2. withdraw();

SBI a3 = new SBI();

a3. Show();

a3. Deposit();

a3. withdraw(); }

Compile-time Polymorphism :-

16th July 2015

The method call will bind with method behaviour at compile time is called compile time polymorphism.

* In compile time polymorphism which method will gets executed will be decided at compile time.

* We can achieve compile time polymorphism by using overloading.

* Compile time polymorphism is also called as static polymorphism or early binding.

Run-time polymorphism :-

The method behaviour will change at runtime i.e. which method ^{will} gets executed will be decided at runtime. is called run time polymorphism

* It is also called as dynamic polymorphism or late binding.

* We can achieve run time polymorphism by using overriding.

Runtime Polymorphism:- The method call will bind with method behaviour at runtime is called as runtime polymorphism. i.e. at compile time the compiler will check base class method and at runtime the derived class method will gets executed.

Rules for Runtime Polymorphism :-

1. Overriding

2. Upcasting (Casting)

Upcasting means creating object for derived class and reference for base class.

Eg:- Using System;

Class A

{

public virtual void show()

{

CWL ("I am A show");

}

Class B:A

{

public override void show()

{

CWL ("I am B show");

}

Class program

{

Static void Main()

{

A a1 = new B();

a1.show();

}

}

Method checking

- at compile time

Method calling

- at runtime.

A a1 = new B(); } *runtime polymorphism.*

Output:- I am B Show.

Q) How to Stop Overriding?

By making the method as sealed.

Eg:- to stop overriding:-

Class A

```
{ public virtual void Show()
  {
    CWL ("I am A Show"); } }
```

Class B:A

```
{ public sealed override void Show()
  {
    CWL ("I am B Show"); } }
```

Class C:B

```
{ public override void Show()
  {
    CWL ("I am C Show"); } }
```

Class program

```
{ static void Main()
  {
    B c1 = new C();
    c1.Show(); } }
```

17th July 2015

Abstract Class :- Abstract class must declare with abstract keyword.

Method = Method heading + Method Body.

Methods are of 3 Types.

1. Static Method.
2. Concrete Method. (Instance Method | Complete Method)
3. Abstract Method.

* Static Method must be declared with static keyword.

* Concrete Method = Method heading + Method Body.

⇒ Method heading will always tell what a method can do.

⇒ Method body will always tell how a method can be implemented.

* Abstract Method will have Only Method heading no method body.

* Abstract Method is also called as incomplete method.

* We can declare both static / instance Variables, methods (static / instance), constructors (static / instance), properties (static / instance), Abstract methods in Abstract class.

* We can declare abstract methods in abstract class and interface

* Abstract method must be declared with abstract keyword.

Syntax :- public abstract void MethodName();

Q) Does Abstract class will participate in inheritance?

Yes

Q) Can we create an object for abstract class?

No we cannot create object for abstract class.

Q) Can we create reference for abstract class?

Yes we can create reference for abstract class.

* whenever a class extends from Abstract class we have to implement the abstract method in derived class.

* we have to override the ~~derived~~ abstract class method in the derived class. (Here abstract is itself "virtual"),

Eg: Using system;

Abstract class A

{

public abstract void show();

}

Class B:A

{

public override void show()

{

CWL("I am show");

}

Class program

{

static void Main()

{

B b1 = new B();

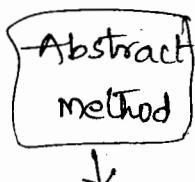
b1.show();

A a1 = new B();

a1.show();

}

,



20th July 2015

May or May not
override in
derived class

Must override
in
DC

Q) Can we declare abstract method in concrete class?

No.

* If we ~~forget~~ ^{are not} ~~do not~~ implementing the abstract method in derived class what to do?

Then make the derived class as abstract class.

Eg:- Using System;

abstract class A

{

public abstract void show();

}

abstract class B:A

{

}

class Program

{

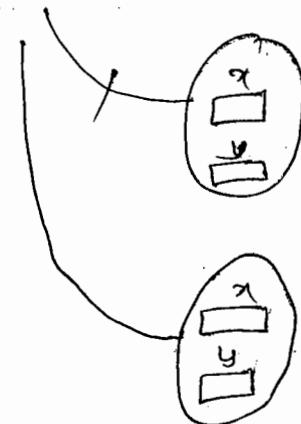
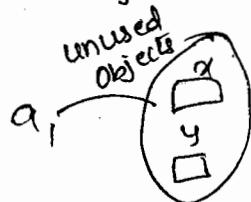
static void Main()

{

}

How to destroy an object?

By making an object as unreferenced (or) ~~decreasing~~ ^{assigning} the ^{object} reference as null.



No. of Objects created = 3

No. of ref. Created = 3

No. of Objects live = 3

|| ref. live = 3

No. of Objects created = 3

" ref. created = 1

" Objects live = 1

" ref. live = 1

* These unreferenced Objects are Collected by garbage collector and destroy the object.

How to destroy an object?

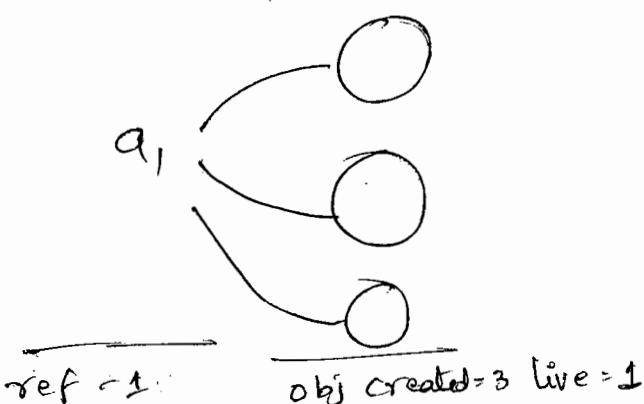
We can destroy the object in 2 ways:

1. By making the object as unreferenced

A a1 = new A();

a1 = New A();

a1 = new A();



2. By assigning null value to the object reference

```
A a1 = new A();
```

```
a1 = null;
```

* Garbage collector will destroy unreferenced objects.

Eg:- for abstract class.

using System;

```
class abstract class ElectricityDept  
{
```

```
    public abstract void calculatePowerBill (int noofunits)
```

```
    public void payBill ()
```

```
    { CWL ("Pay Bill @ eseva, online"); }
```

```
}
```

```
class industry : ElectricityDept
```

```
{
```

```
    public override void calculatePowerBill (int noofunits)
```

```
    { CWL ("Total Bill for industry is " + (no of units * 7)); }
```

```
}
```

```
class CommercialComplex : ElectricityDept
```

```
{  
    public override void calculatePowerBill (int noofunits)
```

```
    { CWL ("Total Bill for CC is " + (no of units * 5)); }
```

```
}
```

```
}
```

Class Residence: Electricity Dept

{

 Public override void calculatePowerBill (int no.of units)

{

 CWL ("Total Bill for Residence" + (no.of units * 3));

}

Class program

{

 Static void Main ()

{

 ElectricityDept el = new Industry();

 el.calculatePowerBill(100);

 el.payBill();

 el = new commercialComplex();

 el.calculatePowerBill(100);

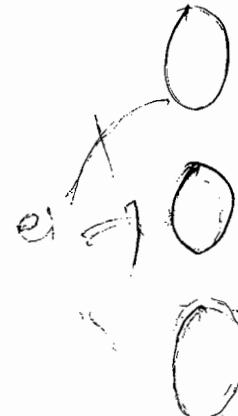
 el.payBill();

 el = new Residence();

 el.calculatePowerBill(100);

 el.payBill();

}



22nd July 2015

Interface :- Interface is a contract or an agreement between itself and its implemented class.

* Interface is used to achieve multiple Inheritance

* Interface consists of only abstract methods.

* We cannot declare variables, methods, constructors in interface.

Q) Can we create an object for interface?

No we cannot create object for interface or abstract class.

Q) Can we create a reference for interface?

Yes we can create reference for an interface.

Q) Does interface will participate in inheritance?

Yes.

Note :- By default interface members are public and abstract

Syntax for Interface :-

==== * =====

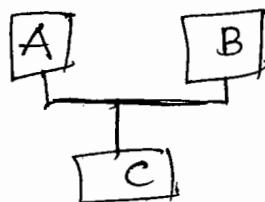
Interface Interfacename

{

}

Q) What is multiple Inheritance?

Creating a derived class by using more than one base class

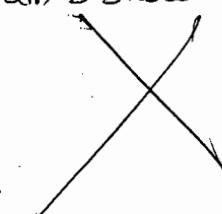


Class A

```
{ Public void show()
  { I am A show
  }
```

Class B

```
{ Public void show()
  { I am B show
  }
```



Class C : A, B

```
{
```

```
}
```

```
C c1 = new C();
```

c₁.show (Ambiguity problem occurs).

to execute which show whether
Class A (or) Class B.

Note:- In the above program whenever we create an object for C class and whenever we access the show method with C class reference the ambiguity problem will occur because at the time of compilation the compiler will confuse that which method must get binded whether A show (or) B show.

* To overcome the above Ambiguity problem we can go for interfaces.

* So, C# .net does not directly support multiple inheritance through classes.

* We can achieve multiple inheritance in C# .net or Java by using Multiple Interfaces.

* Overriding is not possible with interface.*

Eg:- Using System;

Interface A

```
{ Void Show(); }
```

Class B:A

```
{
```

```
Public Void show()
```

```
{ C.WL("I am show"); }
```

```
}
```

Class program

```
{
```

```
Static Void Main()
```

```
{
```

```
B b1=newB();
```

```
b1.show(); }
```

```
}
```

Class A

Class B:A (extends)

Abstract class A

Class B:A (extends)

Interface A

Class B:A (implements)

Interface A

Interface B:A (extends)

Class C:B (implements)

Eg:- Using system;

Interface A

```
{ Void show(); }
```

Interface B:A

```
{ Void Display(); }
```

Class C:A,B

```
{
```

```
Public Void Show()
```

```
{ C.WL("i am show"); }
```

```
Public Void Display()
```

```
{ C.WL(" i am display"); }
```

```
}
```

Class program

```
{
```

```
Static Void Main()
```

```
{ C c1=newC();
```

```
c1.show();
```

```
c1.Display();
```

```
}
```

```
}
```

e.g: Using system;

interface A

{ void show(); }

interface B

{ void show(); }

class C:A,B

{

void A.show()

{ cout("i am A show"); }

void B.show()

{ cout("i am B show"); }

}

class program

{

static void Main()

{ C c1 = new C();

((A)(c1)).show();

((B)(c1)).show();

}

23rd July 2015

* Interface is a contract (or) an agreement between itself and its implemented class.

* Whenever we declare interface it means that we are providing some set of specifications. Any class that is implementing the interface must obey the specifications in the derived class.

What is the use of creating reference for interface (or) abstract class?

By using interface (or) abstract class reference we can access the implemented methods in the derived class.

Eg:- Using System;

interface Shape

{ void draw(); }

Class Triangle : shape

{ public void draw()

{ C.WL ("Draw a Triangle"); }

}

Class Rectangle : shape

{ public void draw()

{ C.WL ("Draw Rectangle"); }

}

Class Circle : shape

{ public void draw()

{ C.WL ("Draw Circle"); }

}

}

class Program

{

static void Main()

{

Shape s = new Triangle();

s.draw();

s = new Rectangle();

s.draw();

s = new Circle();

s.draw();

}

}

24th July 2015

by default
they are public
if abstract

S. No.	Types	class	static	seals	Partial	Abstract	Interface
1.	Can we declare Static Variables in	✓	✓	✓	X	X	X
2.	Can we declare Instance Variables in	✓	✓	✓	X	X	X
3.	Can we declare static methods in	✓	✓	✓	X	X	X
4.	Can we declare Instance Methods in	✓	✓	✓	X	X	X
5.	Can we declare Abstract methods in	✓	✓	✓	X	X	X
6.	Can we declare Static Constructor in	✓	✓	✓	X	X	X
7.	Can we declare Instance Constructor in	✓	✓	✓	X	X	X
8.	Can we declare static properties in	✓	✓	✓	X	X	X
9.	Can we declare Instance Properties in	✓	✓	✓	X	X	X
10.	Can we declare Abstract Properties in	✓	✓	✓	X	X	X
11.	Can we create Object	✓	✓	✓	X	X	X
12.	Can we Create Reference Variable for	✓	✓	✓	X	X	X
13.	Does Inheritance is Possible	✓	✓	✓	X	X	X
14.	Does Multiple Inheritance is Possible	✓	✓	✓	X	X	X
15.	Does Overloading is Possible	✓	✓	✓	X	X	X
16.	Does Overriding is Possible	✓	✓	✓	X	X	X
17.	Does Access Modifiers is	✓	✓	✓	X	X	X

Note :- If the requirement is only Concrete Methods then we have to use Concrete class.

- * If the requirement is only Abstract Methods then we have to use ~~Abstract Class~~ Interface.
- * If the Requirement is both concrete and Abstract Methods then we have to use Abstract class.

25th July 2015

Properties :- Properties are used to transfer the data between the classes

- * Properties are used to set the value and get the value from the private Variables.
- * For every Variable we have to create a property which means for every Variable we have to Create 2 methods Setmethod and getmethod.
- * Set method is used to set the value
- * get method is used to return the value.

Rules to declare the property:-

1. The variablename and propertyname both must be Same
2. The Datatype of the Variable and the Datatype of the Property both must be same.
3. Variable must be declare with lowercase and property must declare with Uppercase character.
4. Variable must declare as private and property must declare as public.

Syntax to declare property :-

Public datatype propertyname

{

Set { variablename = Value; } // Set-accessor / set method

get { return VariableName; } // get-accessor / get method

}

Set method / set accessor / setter block

get method / get accessor / getter block.

Different types of properties are :-

1. Read-only property
2. Write-only property
3. Read-write property
4. Static property
5. Abstract property
6. Automatic properties

* property will gets executed when we call it.

Syntax to call the property :-

Objectname.propertyname;

Properties provide data abstraction

data abstraction (hiding the functionality)

Abstract class provides 0 to 100% data abstraction

Interface provides 100% data abstraction.

eg:- Using system;

Class student

{

int sno;

String sname;

Public int sno

{ set { sno = value; }

get { return sno; }

}

Public String sname

{ set { sname = value; }

get { return sname; }

}

}

Class program

{

static void Main()

{

Student s1 = new Student();

s1.sno = 101;

s1.sname = "Anil";

C.WL (s1.sno);

C.WL (s1.sname);

}

}

27th July 2015

Properties are used to validate the data before setting the values in the variable i.e. we can write validation logic within the properties.

e.g. Using System;

Class Student

{

int sno; string sname; int age;

Public: int sno

{ set { sno = value; }

get { return sno; }

}

Public String sname

{ set { sname = value; }

get { return sname; }

}

Public int age

{ set {

if (value > 0)

{ age = value; }

else

{ age = 0; }

}

get { return age; } } }

Class Program

{

Static void Main()

{

Student s1 = new Student();

CWL ("Enter sno");

s1.sno = int.Parse (C.RL());

CWL ("Enter sname");

s1.sname = C.RL();

CWL ("Enter age");

s1.age = int.Parse (C.RL());

CWL ("sno is " + s1.sno);

CWL ("sname is " + s1.sname);

CWL ("Age is " + s1.age);

} }

Static properties :- Static property must be declared with Static Keyword.

Syntax to access static property is

Classname.PropertyName;

e.g:- Using System

Class A

```
{  
    static String name;  
    public static String Name  
    { set {name=Value;}  
     get {return name;}  
    }  
}
```

Class program

```
{  
    static void Main()  
    { A.Name = "Sathya";  
        C.WL(A.Name);  
    }  
}
```

Automatic properties :- These properties are used to automatically set the values and get the values for the private variables.

* While working with automatic properties, it is not required to declare the variables. At compile time, the compiler will declare the variables by seeing the property names.

Syntax :- Public datatype propertyName
{
 set;
 get;
}

Eg:- Using System

Class A

```
{ public string Name {set; get;}  
    public string Address {get; set;}  
}
```

Class Program

```
{  
    static void Main()  
    {  
        A a1 = new A();  
        a1.Name = "sathya";  
        a1.Address = "Hyd";  
        Console.WriteLine("name is" + a1.Name);  
        Console.WriteLine("Address is" + a1.Address);  
    }  
}
```

Abstract property :- Abstract properties must be declared with Abstract keyword.

* Abstract property will not have body.

* It must be implemented in the derived class.

Abstract property must be declared in abstract class only.

Eg:- Using System;

abstract class A

```
{ public abstract string Name {get;} }
```

Class B:A

```
{ public override string Name  
{  
    get { return "Sathya Tech"; }  
}
```

Class program

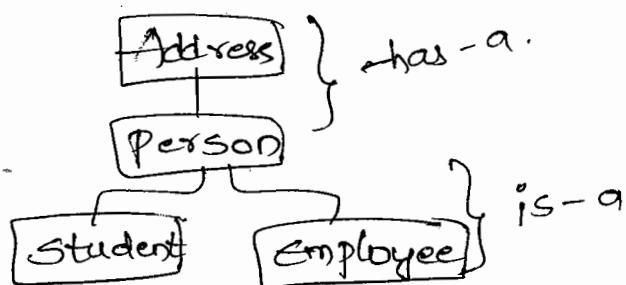
{

Static Void Main()

{

B b1 = new B();

C. wL (b1. Name. ToString());



eg for has-a relationship.

Has-a relationship :- If we want to use the object more than once then we have to use has-a relationship.

Q Using system;

Class Address

{

Public String Dno {get; set;}

public String sname {get; set;}

public String cname {get; set;}

}

Class person

{

Public String Fname {get; set;}

public String Lname {get; set;}

Public Address TempAddress {get; set;}

Public Address PermanentAddress {get; set;}

}

Class Program

{

Static Void Main()

{

Person P = new Person();

~~P =~~

P.TempAddress = new Address();

P.permanentAddress = new Address();

P.Fname = "Anil";

P.Lname = "Kumar";

P.TempAddress.Dno = "1-2-3";

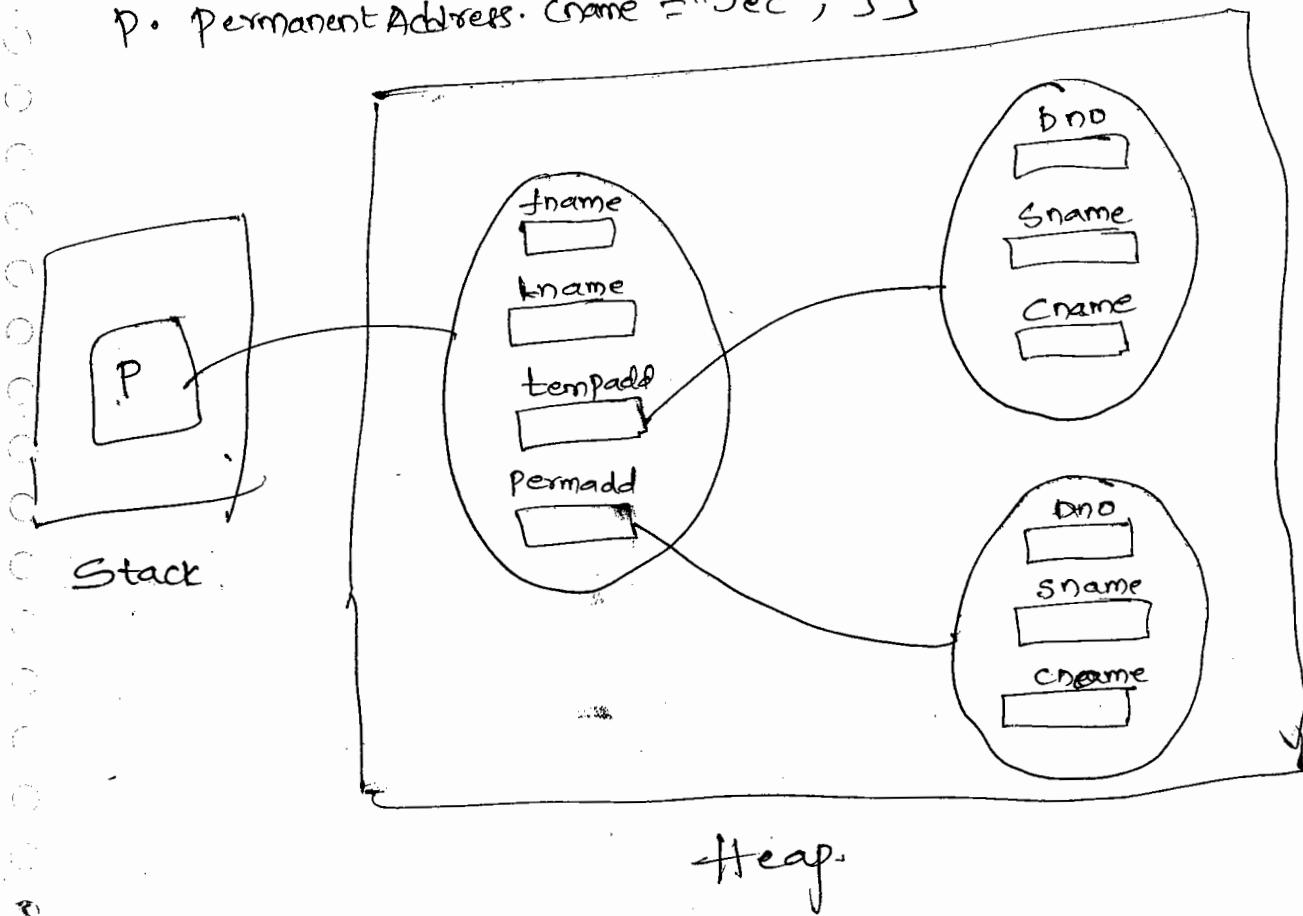
P.TempAddress.Sname = "Ameerpet";

P.TempAddress.Cname = "Hyd";

P.permanentAddress.Dno = "4-5-6";

P.permanentAddress.Sname = "Malkygen";

P.permanentAddress.Cname = "Sec"; } }



Assemblies

[26th July 2015]

Assembly :- Assembly is the compiled format of any .net program which may be .dll or .exe

- * .exe is executable file but not reusable.
- * .dll is reusable file but not executable.



Whenever we compile any c# .net program, the compiler will check for syntax errors and it will generate either .exe or .dll

Console App → .exe

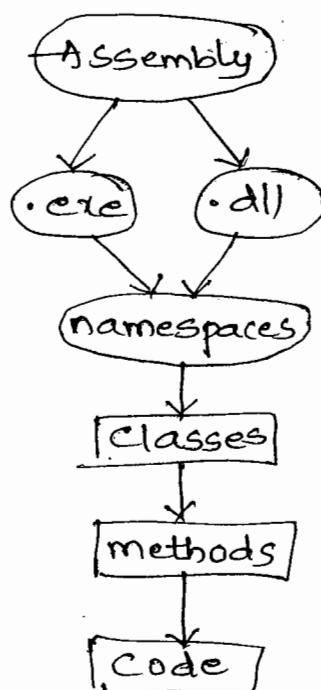
Windowsforms App → .exe

WPF App → .exe

Web APP → .dll

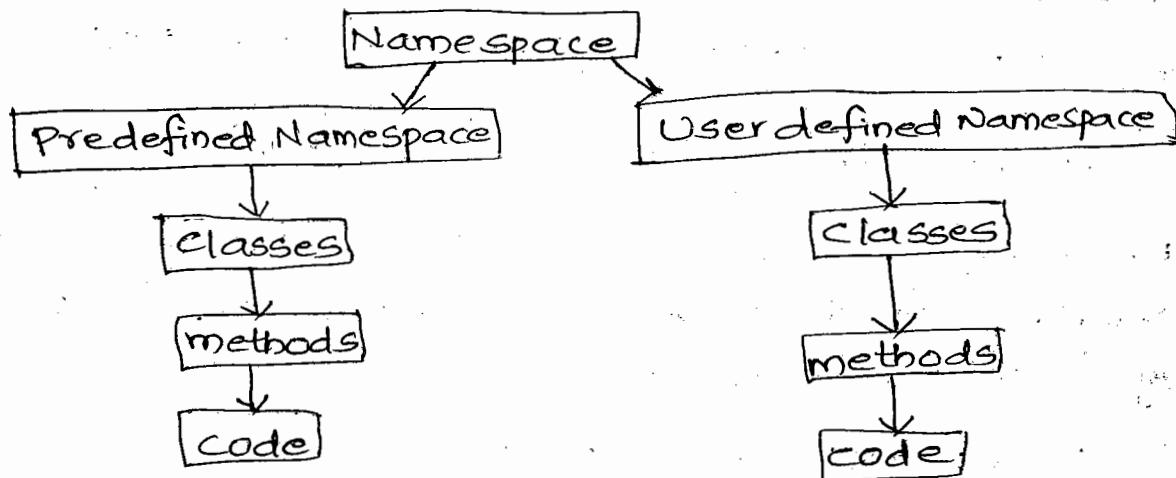
WCF APP → .dll

Class Library APP → .dll



What is the difference b/w Assembly and namespace?
Assembly is collection of namespaces and
namespace is collection of classes.

Namespaces are of two types



- * Predefined namespaces was written by microsoft
- * Userdefined namespaces was written by programmer depending on user requirement.

Syntax for declaring namespace:-

```
namespace namespacename
{
    class classname
    {
        void methodname()
        {
        }
    }
}
```

KMa
{
C1
{
C1
{
C1
}
na:
{
ca
clu
}

eg:- namespace System

{

Class console

{

Static void WriteLine()

{

}

{

}

}

* If we want to consume the predefined classes or methods we have to use the namespace.

Syntax to use the namespace is

using namespacesname;

eg:- using System;

using System.Data;

using System.Data.SqlClient;

using System.IO;

using System.Linq;

using System.Mvc;

Namespace n1

{

Class A

{ }

Class B

{ }

}

Namespace n2

{

Class C { }

Class D { }

}

A.cs → CSC A.cs → A.exe

- Assemblies are of three types
1. Private Assembly
 2. Public Assembly (or) Shared Assembly
 3. Satellite Assembly.

Public - anyone	P.
private - within class	{
internal - within assembly	

Private Assembly :- The Assembly that was specific for a single Application or a specific Folder is called as private Assembly or Folder Specific Assembly.

* We can create dll file in class library template.

Example to. Create a dll and consume in Console Appn :-

Step-1 :- go to E Drive and create a Folder with name Demoassembly.

Step-2 :- goto ---> File ---> New ---> project ---> Select class Library Template. --->
name = mydll --->
Browse = E:\demoassembly ---> OK

Step-3 :-

```

using System;
using System.Data.SqlClient;
namespace mydll
{

```

public class MySqlClass

{

SqlConnection GetConnection()

{

return new SqlConnection("User id =sa; Password=abc;
database = Sathya; data source = KANNABANU-PC")

Step

```
class Employee
```

```
{
```

```
    SqlConnection con = GetConnection();  
    con.Open();
```

```
    String query = "insert into emp values ('" + eno + "  
        " + ename + "', '" + salary + "')";
```

```
    SqlCommand cmd = new SqlCommand(query, con);
```

```
    int i = cmd.ExecuteNonQuery();
```

```
    con.Close();
```

```
    return i;
```

```
}
```

```
Public int DeleteEmployee (int empno)
```

```
{
```

```
    SqlConnection con = GetConnection();
```

```
    con.Open();
```

```
    String query = "delete from emp where eno=" + empno;
```

```
    SqlCommand cmd = new SqlCommand(query, con);
```

```
    int i = cmd.ExecuteNonQuery();
```

```
    con.Close();
```

```
    return i;
```

```
}
```

```
}
```

```
}
```

Step-4 :- build → buildSolution → OK

whenever we compile the class library project the compiler will generate .dll file go and check in

E:\demoassembly\mydll\bin\debug\mydll.dll

Step-5 :- Consume the dll file in console App

Step-6 :- goto ---> Start ---> run ---> devenv ---> OK

File ---> New ---> Project ---> Select ConsoleApp ---> OK

go to Solution Explorer ---> References ---> Add Reference ---> Select Browse ---> go to

E:\demoassembly\mydll\bin\debug\mydll.dll

Select mydll.dll ---> OK

Step-7 :- Write the code:-

```
using System;
using mydll;
namespace ConsoleApplication1
{ class Program
    { static void Main()
        {
```

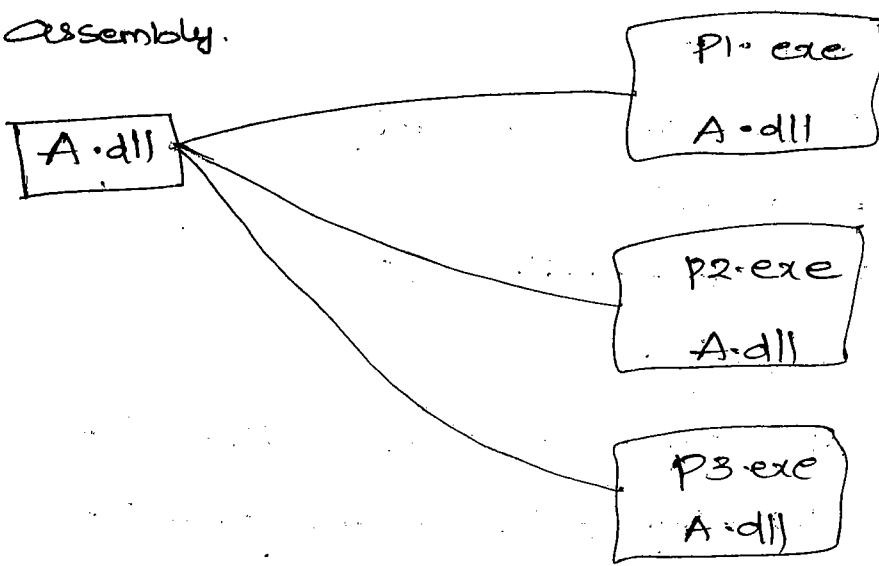
```
        mydll.Mysqlclass obj = new Mysqlclass();
    }
```

```
    C.WL("Enter your choice");
    C.WL("1. Insert");
    C.WL("2. Delete");
    int choice = int.Parse(Console.ReadLine());
```

```
if (choice == 1)
{ C-WL ("Enter Empno");
int eno = int.Parse (console.ReadLine ());
C-WL ("Enter Empname");
String ename = console.ReadLine ();
C-WL ("Enter Salary");
double salary = double.Parse (console.ReadLine ());
int i = obj.CreateEmployee (eno, ename, salary);
if (i == 1)
{ C-WL ("Record is inserted"); }
else
C-WL ("Insertion failed"); }

else
{ C-WL ("Enter Empno");
int eno = int.Parse (console.ReadLine ());
int i = obj.DeleteEmployee (eno);
if (i == 1)
{
C-WL ("Record is deleted"); }
else
C-WL ("Deletion Failed"); }
```

* Private assembly is also called as folder specific assembly.



* Whenever we consume the dll file then a separate copy of the dll file is available in each and every folder. So, private assembly is also called as folder specific assembly.

Public assembly :- The assembly that was registered under GAC Location is called as Public Assembly.

(Q) What is GAC?

Global Assembly Cache

* GAC is a common place where we have to place the dll file once the assembly or dll was registered under GAC a separate copy of dll file will not be available in a specific folder.

* All predefined dlls are public assemblies.

(Q) Where is GAC location?

C:\Windows\Microsoft.NET\Assembly\GAC-MSTL

* I
we
key
* pi
giv
eg
Step -

using
name
{
publ
{
pl
{
c.
3

Step -
goto
Do
Si
Cho
--
derr

* In order to register the dll file in GAC location we have to create a strongname and public key token.

* Public key token is a unique identification no given for the dll file.

Eg for public dll :-

Step-1 :- goto ... → File → New project → select Class library → name = demopublicdll
Location = E:\demodassembly → OK.

```
Using System;
namespace demopublicdll
{
    public class A
    {
        public void show()
        {
            Console.WriteLine("I am show");
        }
    }
}
```

Step-2 Create a strongname.

goto → Project on the menubar → demopublicdll Properties → Signing
Sign the assembly.
Choose Strongname → select dropdown → New
→ Strongname = mykey → Build → Build Solution
demopublicdll.dll is generated.

Step-3:- In order to register dll file in GAC location
we have to use gacutil tool.

GACUTIL TOOL is a tool which is used to register
dll file under GAC location.

Step-4:- Open Visual Studio command prompt

gacutil -i E:\demoadm\demopublic.dll\bin
Assembly added successfully Debug.

Step-5:- go and check whether dll file is registered
under GAC.

C:\Windows\Microsoft.NET\assembly\GAC_MSIL
Demopublic.dll

Step-6:- Consume the dll in console app

goto ---> Solution Explorer ---> &c on References --->
Add Reference ---> Browse --->
C:\Windows\Microsoft.NET\assembly\GAC_MSIL
demopublic.dll\V4.0-1.0\demopublic.dll.dll

Step-7:- using System;

using demopublic.dll;

namespace ConsoleApplication2

{

Class program

{

Static void Main()

{

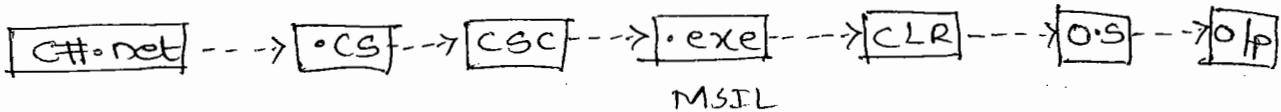
demopublic.dll.A Obj = new A();

Obj.Show(); } }

Satellite Assembly :- The assembly that was developed in multiple languages is called as satellite assembly.

Exception Handling

1st Aug 2015



* Every C#-net program must save with .CS extension.

* Whenever we compile the program the Compiler will check for Syntax errors which are also called as Compile time errors.

Q) What are Compile time Errors ?
Syntax errors.

Q) When the Compile time errors will occur ?
At the time of compilation of the program

Q) Can we rectify compile time errors ?
Yes.

Q) Who will identify compile time errors ?
Compiler.

Q) Who will rectify compile time errors ?
Programmers.

Q) Why compile errors will occur ?

These errors will occur because of the invalid Syntax that was written by programmers.

Q) What is Exception?

Exception is a runtime error.

Q) When Exception will occur?

At the time of execution of the program.

Q) Who will identify run time errors?

CLR.

Q) Can we rectify run time errors?

No we cannot rectify run time errors but we can handle run time errors.

Q) What is the problem with runtime errors?

Abnormal Termination of the Program

Q) Can we handle run time errors?

We can handle Runtime Errors by using Exception Handling Mechanism.

Q) What is Exception Handling?

Exception Handling Mechanism is a Mechanism which is used to handle Runtime Errors.

Eg:- Using System;

Class Demo

{

SVM ()

{

C·WL ("Enter a no");

int a = int.parse(C·RLC());

C·WL ("Enter b no");

int b = int.parse(C·RLC());

int c = a/b;

C·WL ("Quotient is" + c); } }

Observation :- When we execute the above program if the user gives the input as 10 and 0 then an exception will occur saying that "Attempted to Divide by Zero."

* We can handle the Exception in 2 ways

1. Logical Implementation
2. try-catch implementation

Logical Implementation :- The programmer must analyze that what type of Exception will occur and he has to write the logic to handle the exception.

Eg :- Handling the runtime error by using logical implementation

using System;

Class A

{

SMCL

{

C.WL ("Enter a no");

int a = int.parse (C.RLC());

C.WL ("Enter b no");

int b = int.parse (C.RLC());

if (b == 0)

{ C.WL ("Denominator must not be 0"); }

else

{ int c = a/b;

C.WL ("Quotient is" + c);

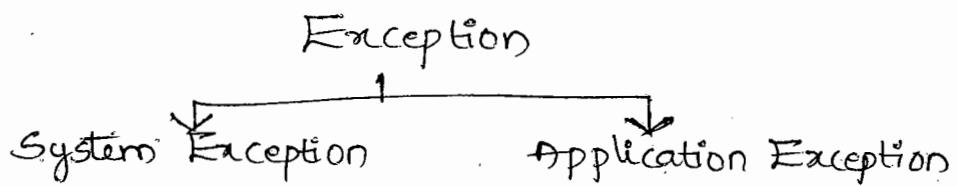
}

{

?

* It is difficult for the programmer to identify the error and to write the logic for the error.

* So Microsoft has given try-catch implementation to handle Runtime Errors.



System Exceptions :- System Exceptions are pre-defined Exceptions that can be handled by CLR.

* In .NET all Exceptions are predefined classes which are inheriting from Exception class.

Different types of System Exception :-

System Exception
↓

Divide By Zero Exception

FormatException

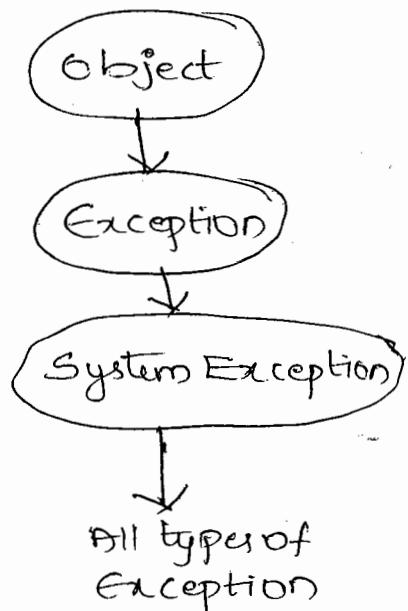
IndexOutOfRangeException

SqlException

StackOverflowException

OverflowException

InvalidCastException



The try-catch Exception :-

try - catch - finally

try

{

We need to write code and whenever an Exception will occur the control goes to catch block

}

catch (Exception e)

{

We need to display error message in catch block, Catch block will handle exception and display the error message

}

finally

{

The code that we write inside finally block will gets executed even if exception occurs

}

try

{

1 ← Exception was occurred ---> CLR will identify what
2 type of Exception was occurred and CLR will create
3 Object for the corresponding Exception class and
4 catch (Exception e) Control goes to catch block.

{

e.WL(e.Message); → in catch block object is for Derived
class and Reference is for Exception class,
so the overridden property will gets executed.

finally

{

}

Eg:- for real time eg for Dynamic Polymorphism
Overriding.

eg

Class Exception

{

Public Virtual String Message

{
get;

}

}

Class DivideByZeroException: Exception

{

Public override string message

{
get

{

return "Attempted to DByZero";

} } }

Class FormatException: Exception

{

Public override string message

{
get

{

return "I/P string was not in a correct format";

} } }

Class OverflowException: Exception

{

Public override string message

{
get

{

return "Value to be small or big";

} } }

*

OCC

IFC

Divide

will

class

(or)

* It

pre

to

eg:- using System;

```
class A
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Enter a no");
            int a = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter b no");
            int b = int.Parse(Console.ReadLine());
            int c = a/b;
            Console.WriteLine("Quotient is "+c);
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
        finally
        {
            Console.WriteLine("I will execute even if exception occurs");
        }
    }
}
```

* In the above program whenever exception will occur (if DivideByZero exception will occur), i.e if denominator is zero, then CLR will identify that divide by zero exception was encountered and CLR will create an object for divide by zero exception class and it will print the overridden message (or) predefined message.

* It is difficult for the user to ~~identify~~ understand predefined error messages so the programmer has to display user friendly messages to the users

eg :- Using System;

Class A :

{

SVM()

{

try

```
{ C-WL ("Enter a no");
int a = int.parse (C-RLC));
C-WL ("Enter b no");
int b = int.parse (C-RLC);
int c = a/b;
C-WL ("Quotient is" + c);
```

}

Catch (DivideByZeroException)

```
{ C-WL ("Denominator must not be 0"); }
```

Catch (FormatException)

```
{ C-WL ("Please enter no"); }
```

Catch (Exception e)

```
{ C-WL (e.message); }
```

Only try - Invalid

only catch - Invalid

Only finally - Invalid

try-catch - Valid

try-finally - Valid

try-catch-finally - Valid

try-try-catch-finally - Valid

A

=

Tb

use

Eg

C

Ck

{

S

{

t-

{

?

Ca

{

th-

obj

Application Exception :-

The Exception that was created depending on the user requirement is called as Application Exception.

Eg:- Using System;

Class EvennumberException : Exception

{

 public override string Message

{
 get
 {

 return "please enter Even no";

 }
}

Class Program

{

 SVMC)

{

try

{ C·WL ("Enter Even no");

 int no = int.parse (C·RL());

 if (no%2 == 0)

 { C·WL ("Even no"); }

 else

 { throw new EvennumberException(); }

}

catch (Exception e)

 { C·WL (e·Message); }

throw Keyword is used to throw the Exception object to CLR

Delegates

25th Aug 2015

Delegate :- Delegate is a type which hold the information of the method.

* Delegate is a reference pointer for a method i.e. with delegate reference we can call methods.

* Delegate is a Safe type function pointer which points to method.

* By using delegates we can pass methodname as parameter.

* Delegates are used to develop Event Driven Programming.

* At compile time delegate will be converted to class.

* By using delegate reference we can call multiple methods.

Steps to work with delegate :-

1. Create a delegate

Syntax :- access modifier delegate returntype delegatename();

e.g:- Public delegate void Mydelegate (Parameters);

2. Create a method for delegate

Public void Show()

{

}

3. Create an object for the delegate and Pass the method name as Parameter.

Mydelegate Objectname= new Mydelegate (Show);

4. Invoke the delegate object

When we invoke the delegate object then delegate will invoke the methods.

```
delegate Objectname();
```

* At the time of invoking delegate object we have to pass values.

* The number, order, type of values that we pass must match with number, order, type of parameters of delegate.

Rules to declare delegate :-

1. The returntype of delegate and the returntype of method both must be same.

2. The Parameters of delegate and the Parameters of method must be same.

e.g:- Using System;

// Create delegate

```
public delegate void MyDelegate();
```

Class program

```
{
```

// Create a method for the delegate

```
Static void Show()
```

```
{
```

```
Console.WriteLine("I am Show");
```

```
}
```

```
Sum()
```

```
{
```

// Create an object for the delegate and pass the Method name as Parameters.

```
MyDelegate Obj = new MyDelegate(Show);
```

// invoke the delegate object

```
obj();
}
}
```

Eg:- using System;

```
public delegate void MyDelegate();
```

class Program

```
{
```

```
static void Show()
```

```
{
```

```
CWL ("I am show");
```

```
}
```

```
static void Display()
```

```
{
```

```
CWL ("I am Display"); }
```

```
static void point()
```

```
{
```

```
CWL ("I am point"); }
```

```
static void Main()
```

```
{
```

```
MyDelegate = new MyDelegate (Show);
```

```
Obj = Obj + new MyDelegate (Display);
```

```
Obj+ = new MyDelegate (Point);
```

```
Obj();
```

```
}
```

```
}
```

Example with Delegate with parameters.

```
using System;
public delegate void MyDelegate (int x, int y);
class Program
{
    static void Add (int x, int y)
    {
        Console.WriteLine ("sum is" + (x+y));
    }
    static void Sub (int x, int y)
    {
        Console.WriteLine ("Diff is" + (x-y));
    }
    static void Mul (int x, int y)
    {
        Console.WriteLine ("Product is" + (x * y));
    }
    static void Main()
    {
        MyDelegate obj = new MyDelegate (Add);
        obj += new MyDelegate (Sub);
        obj += new MyDelegate (Mul);
        obj (30, 20);
    }
}
```

Delegates with return type with parameters:

using System;

public delegate int MyDelegate (int x, int y);

Class program

{

Static int Add (int x, int y)

{

return x+y;

}

Static int sub (int x, int y)

{

return x-y;

}

Static void Main()

{

My delegate obj = new MyDelegate (Add);

int sum = obj(30, 10);

obj += new MyDelegate (sub);

obj(30, 10);

int diff = obj(30, 10);

CWL(result);

}

}

*

Eve

to

exe

*

Who

Met

have

Step

1.

P

2. (

Syn

3. (

)

4.

5.

Ego

* Delegate will invoke all the methods

[Event → Delegate → methods]

Event: Events are the time periods which intimate to the delegate that which method must get executed.

* Events are the members of the class

What is the difference b/w method and Event?

Method will have return type but Event does not have any return type.

Steps to create Event:-

1. Create a delegate

public delegate void MyDelegate();

2. Create an event for the delegate

Syn:- [public event Delegatename eventname;]

3. Create a method for delegate

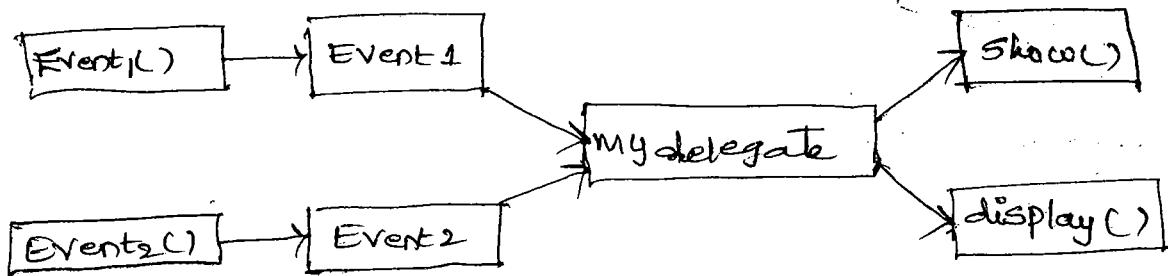
```
public void Methodname()
{
}
```

4. Add the delegate object to event

eventname += new MyDelegate (methodname);

5. Call the event.

Eg:- eventname();



```
using System;
Public delegate void MyDelegate();
class A
{
    Public static event MyDelegate Event1;
    Public static event MyDelegate Event2;
    Static void Show()
    {
        CWL ("I am show");
    }
    Static void Display()
    {
        CWL ("I am Display");
    }
    Static void Main()
    {
        Event1 += new MyDelegate (Show);
        Event2 += new MyDelegate (Display);
        Event1();
        Event2();
    }
}
```

To develop a form

```
using System;
using System.Windows.Forms;
class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

```
private void initializeComponent()
```

```
{
```

```
this.Name = "Myform";  
this.Text = "Demoform";  
this.Height = 200;  
this.Width = 300;
```

```
this.Load += new EventHandler(Form1_Load);  
this.Click += new EventHandler(Form1_Click);  
}
```

```
private void Form1_Load(object sender, EventArgs e)
```

```
{
```

```
MessageBox.Show("I am Form Load method");
```

```
}
```

```
private void Form1_Click(object sender, EventArgs e)
```

```
{
```

```
MessageBox.Show("I am form click method");
```

```
}
```

```
static void Main()
```

```
{
```

```
Application.Run(new Form1());
```

```
} }
```

```
= 0 =
```

```
public event EventHandler Load;
```

```
public event EventHandler Click;
```

```
public string Name { set, get; }
```

```
public string Text { set, get; }
```

```
public int Height { set, get; }
```

```
public int Width { set, get; }
```

GH
k

Access Modifiers

28th July 2015

Access Modifiers are used to specify the accessibility permissions for the types and its members.

* Accessibility means who can access and who cannot access.

Different types of Access Modifiers are

1. private
2. protected
3. internal
4. protected internal
5. public

1. private :- The scope of private is within the class
* By default the members of the class are private.

* If we declare variable as private we cannot access the variable outside the class.

* If we declare method as private we cannot access the method outside the class.

* If we declare constructor as private we cannot create object outside the class.

2. protected :- The scope of protected is within the class and ~~for the immediate~~ and the class that is extending from that class i.e. in the immediate derived class.

Rules: 1. Inheritance

2. Reference and Object for derived class.

Eg:- Using System

Class A

{

Protected void show()

{ C.WL ("I am show"); }

}

Class B:A

{

SVML)

{

B b1 = new B();

b1.show(); ✓ Valid

A a1 = new A();
a1.show();

✗ Not valid

}

Rules to access protected members :-

1. Inheritance must be there.

2. Object and reference must be for derived class

Internal :- The scope of internal is within the assembly.

* Internal will work like public within the assembly
and private outside the assembly.

* The default access modifier for any type is
private.

pro

the

A.dll

```
namespace n1
{
    internal class A
    {
        public
        within
        assembly
    }
    Class B
    {
        A a1 = new A();
    }
}
```

✓ (valid)

B.exe

```
Using n1;
namespace n2
{
    class program
    {
        SVM()
    }
    n1.A a1 = new A();
}
```

X (not valid)

Protected Internal :- It will work like public within the assembly and protected outside the assembly.

A.dll

```
namespace n1
{
    class A
    {
        protected internal
        void show()
    }
}

namespace n2
{
    class B
    {
        A a1 = new A();
        a1.show();
    }
}
```

B.exe

```
namespace n3
{
    class C:A
    {
        A a1 = new A(); X
        C c1 = new C();
        c1.show();
    }
}
```

Public :- The scope of public is no restrictions.
i.e anyone can access.

* The scope of public is within the assembly or outside the assembly.

~~Case-1~~ Case-1 :- Using system;

namespace n1

{

Class A

{

Private void Method1()

{ C.WL ("I am private method"); }

Protected void Method2()

{ C.WL ("I am protected method"); }

~~Internal~~ Internal void Method3()

{ C.WL ("I am Internal method"); }

Protected Internal void Method4()

{ C.WL ("I am protected Internal method"); }

Public void Method5()

{

C.WL ("I am Public method"); }

}

Static void Main()

{ A a1 = new A();

a1.Method1();

a2.Method2();

a3.Method3();

a4.Method4();

a5.Method5();

Case-2

c

e

s

o

u

l

l

l

l

l

l

l

l

l

l

l

l

l

l

l

l

l

l

ns. Case-2:- class sum in separate class.

class B
{
sum >
{
A a1=new A();
a1.Method3();
a2.Method4();
a3.Method5();
}
}

Case-3:- with Inheritance

class B:A
{
static void Main()
{
B b1=new B();
b1.Method2();
b1.Method3();
b1.Method4();
b1.Method5();
}
}

Type	private	Protected	Internal	Protected Internal	Public.
1. Within the class	✓		✓	✓	✓
2. Within derived class baseclass object	X	✓	✓	✓	✓
3. Within derived class derived class object	X	✓	✓	✓	X
4. Within the Assembly	X	X	✓	✓	✓
5. Outside the Assembly	X	X	X	X	
6. Outside Assembly derived class object					

Ans

Term

* A

data

* A

Cont

* B

Sor

D.

II.

III.

IV.

V.

VI.

VII.

VIII.

IX.

X.

Arrays

31st July 2015

Arrays:- Array is used to hold multiple values temporarily until the program is running.

* Array is a userdefined reference type datatype which is used to hold multiple values of same datatype in a single variable.

* Array is used to allocate the memory in a continuous memory location i.e side by side.

A[]



* By using arrays we can perform searching and sorting operations easily.

Different Types of Arrays are :-

1. Single Dimensional Array

2. Multi Dimensional Array

3. Jagged Array.

1. Single Dimensional Array:- It is used to store multiple values of same datatype in a single variable.

Syntax :-

datatype [] arrayname = new datatype [size];

Syntax to store the Value in the array :-

arrayname [index] = Value;

* In Array each and every item can be identified by using index no.

* Always the index no starts from 0 to size-1.

Syntax to retrieve the value from the Array:-

Arrayname [Index]

e.g. using System;

Class Program

{

Static void Main()

{

int [] Ar = new int [5];

Ar[0] = 10;

Ar[1] = 20;

Ar[2] = 30;

Ar[3] = 40;

Ar[4] = 50;

for (int i=0; i<5; i++)

C.WL (Ar[i]);

}

}

(b) WAP to find the sum of the numbers in the Array.

Sum()

{

int [] Ar = new int [5];

Ar[0] = 10;

Ar[1] = 20;

Ar[2] = 30;

Ar[3] = 40;

Ar[4] = 50;

int Sum=0;

for (int i=0; i<5; i++)

{

Sum = Sum + Ar[i];

}

C.WL (Sum);

(a) W
find
us
cl
{
stc
{
ir

fied
g) WAP to declare integer array with size 6 and find the sum of even no's and odd numbers.

using System;

Class program

```
{  
    static void Main()  
    {  
        int [] Ar = new int[6]  
        Ar[0] = 10; Ar[1] = 15; Ar[2] = 20; Ar[3] = 25;  
        Ar[4] = 30; Ar[5] = 35; Ar[6] = 40  
        int evensum = 0; int oddsum = 0;  
        for (int i = 0; i <= 5; i++)  
        {  
            if (Ar[i] % 2 == 0)  
            {  
                evensum = evensum + Ar[i];  
            }  
            else  
            {  
                oddsum = oddsum + Ar[i];  
            }  
        }  
        Console.WriteLine("Sum of even no's is " + evensum);  
        Console.WriteLine("Sum of odd no's is " + oddsum);  
    }  
}
```

Syntax :- to declare Single Dimensional Array :-

datatype [] arrayname = new datatype [size] {values};

WAP to count the no of even no's and odd no's in an array.

Using System;

class A

{

 SVMC

{

 int [] Ar = new int [7] { 2, 3, 4, 5, 6, 7, 8 };

 int evenCount=0; int oddCount=0;

 for (int i=0; i<7; i++)

 {

 if (Ar[i] % 2 == 0)

 {

 evenCount = evenCount + 1;

 }

 else

 {

 oddCount = oddCount + 1;

 }

 }

 CWL ("no. of even no's are" + evenCount);

 CWL ("no. of odd no's are" + oddCount);

 }

 }

Eg:-

na

{

ck

{

st

{

co

C

C

do

co

int

int

stor

in

do

do

fr

{

ir

{

C

pr

(

C

C

F

3rd Aug 2015

Eg:- Using System;

namespace ConsoleApplication10

{

Class Program

{

Static void Main()

{

C·WL ("=====");

C·WL ("WELCOME TO TIPSY TOPSY BAKERY");

C·WL ("=====");

double totalBill = 0;

C·WL ("Enter the no of items you want to buy");

int no = int·parse (C·RLC);

int [] ino = new int [no];

String [] iname = new String [no];

int [] q_ty = new int [no];

double [] price = new double [no];

double [] total = new double [no];

for (int i=0; i<=no-1; i++)

{ C·WL ("Enter " + no);

ino [i] = int·parse (C·RLC);

C·WL ("Enter " + iname [i]);

iname [i] = console·RLC();

C·WL ("Enter no of " + iname [i]);

q_ty [i] = int·parse (C·RLC());

C·WL ("Enter the price of " + iname [i]);

price [i] = double·parse (C·RLC());

```

total[i] = qty[i] * price[i];
totalbill = totalbill + total[i];
}

C.wL("Bill is");
C.wL("ino" + "It" + "iname" + "It" + "qty" + "It" + "price" +
     "It" + "total");

for (int i=0; i<=no-1; i++)
{
    C.w(ino[i] + "It");
    C.w(iname[i] + "It");
    C.w(qty[i] + "It");
    C.w(price[i] + "It");
    C.w(total[i] + "It");
    C.wL();
}

C.wL("====");
C.wL("Total Bill is' + totalbill);
C.wL("====");
}
}

```

Output

	ino	iname	qty	price	total
0	VPIZZA	3	60	180	
1	CPIZZA	2	80	160	
2	VBurger	3	120	360	

Total Bill is 700.

Mu

Sy

Dat

[r

Sy

Arr

e.g:-

Multi-dimensional Array (or) 2-dimensional Array:-

Syntax to declare multi-dimensional array:-

Datatype [,] Arrayname = new Datatype

[no of rows, no of columns];

Syntax to store the value in the array is

Arrayname [rowindex, Colindex] = Value;

e.g:- Using System;

Class program.

{

Static void Main()

{

int [,] Ar = new int [2,3];

Ar[0,0]=10; Ar[1,0]=40;

Ar[0,1]=20; Ar[1,1]=50;

Ar[0,2]=30; Ar[1,2]=60;

for(int i=0; i<1; i++)

{

for(int j=0; j<2; j++)

{

C.WL(Ar[i,j] + " It");

}

C.WL();

} }

Syntax :-

datatype[] Arrayname = new datatype [no of rows, no of cols]
 { { row1 values }, { row2 values } } ;

Eg:- int [] Ar = new int [2,3] { {10,20,30}, {40,50,60} } ;

Jagged Array :-

Jagged Array means Array inside Array is called as Jagged Array.

Syntax for Jagged Array :-

datatype [][] Array name = new datatype [no of rows][] ;

Eg:- Using System;

Class A

{

 SVM()

{ int [][] Ar = new int [4][] ;

 Ar[0] = new int [1] {10} ;

 Ar[1] = new int [2] {20,30} ;

 Ar[2] = new int [3] {40,50,60} ;

 Ar[3] = new int [4] {70,80,90,100} ;

 for (int i=0; i<4; i++)

 {

 for (int j=0; j<=Ar[i].Length-1; j++)

 { C.W(Ar[i][j] + " ") ; }

 C.WL();

} }

```
char[][] Ar = new char[5];
```

```
> f()
eg};  
{;  
allied  
;  
for (int i=0; i<=4; i++)  
{  
    for (int j=0; j<=Ar[i].length-1; j++)  
    { c.w [Ar[i][j] + " "); }  
    C.WL();  
}
```

Output

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

Eg: To print the given string in reverse.

```
String s="satyam";  
char[] Ar = s.toCharArray();  
for (int i=5; i>0; i--)  
{  
    c.w (Ar[i]);  
}  
C.WL();
```

Collections

4th Aug 2015

Collections are used to implement data structures in .Net.

What is Data Structure?

Data structure is used to store and manipulate data.

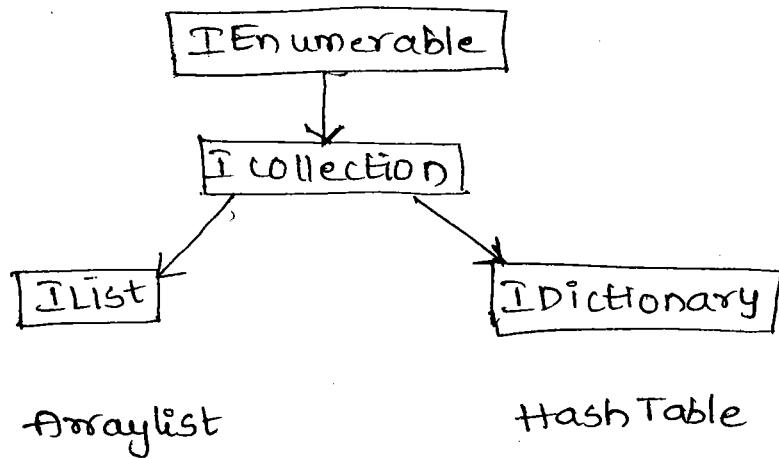
* If we want to perform operations on group of objects then we have to go for collections.

* In C# .net every collection is a predefined class but every class is not a collection.

* In order to work with collections in C# .net, Microsoft has given a predefined namespace.

System.Collections; which consists of predefined classes like

- * ArrayList
- * Stack
- * Queue
- * HashTable
- * Sorted List



es

```
interface IEnumerable
{
    IEnumerator GetEnumerator();
}
```

data
of
class
reson
ined

```
interface IEnumerator
{
    object Current {get;}
    bool MoveNext();
    void Reset();
}
```

```
class ArrayList : IEnumerable
{
    public IEnumerator GetEnumerator()
    {
        return (IEnumerator) this;
    }
}
```

* GetEnumerator() will return the object of the class which is implementing IEnumerable interface.

```
ArrayList Ar = new ArrayList();
IEnumerator ie = Ar.GetEnumerator();
bool b = ie.MoveNext();
Object o = ie.Current;
ie.Reset();
```

6th Aug 2015

Array

- * The size of Array is fixed.
- * We cannot increase (or) decrease the size of the Array depending on the requirement.
- * We cannot insert/remove the items from the Array at a specific position.
- * Array is used to store homogeneous values.
- * Array is used to store multiple values of same datatype.
- * Readymade method Support is not available in Array.
- * Array is not growable.

ArrayList

- * The size of ArrayList is not fixed.
- * We can increase or decrease the size of the ArrayList depending on the requirement.
- * We can insert/remove the items from ArrayList.
- * ArrayList is used to store heterogeneous values.
- * ArrayList is used to store multiple values of different datatypes.
- * Readymade method Support is available in ArrayList.
- * ArrayList is growable.

7th Aug 2015

ArrayList:- ArrayList is a collection class which is used to perform operations on group of objects.

Class ArrayList : IEnumerable

```
{  
    Public Virtual int Add(object value);  
    Public Virtual void Remove (object obj);  
    Public Virtual void RemoveAt(int index);
```

Publi

publi

Publi

Publi

}

1. Ad =
the

2. Re =
item

3. Re =
item

4. In =
in th

5. Re =
item

6. cl =
from

7. Si =
the

Eg:

```
    public virtual void Insert (int index, object value);  
    public virtual void Reverse();  
    public virtual void Clear();  
    public virtual void Sort();  
}
```

- Y
of
ing
.
ove
list
o
values.
If
port
e.
his
ite
- 1. Add() :- This method is used to add the items in the ArrayList.
 - 2. Remove() :- This method is used to remove the items from the ArrayList.
 - 3. RemoveAt() :- This method is used to remove the items from the ArrayList at a specific position.
 - 4. Insert() :- This method is used to add the items in the ArrayList at a specific position.
 - 5. Reverse() :- This method is used to reverse the items in the ArrayList.
 - 6. Clear() :- This method is used to clear the items from the ArrayList.
 - 7. Sort() :- This method is used to sort the items in the ArrayList.

Eg:- interface IEnumerable

```
{  
    Ienumerator GetEnumerator();  
}
```

```
interface Ienumerator  
{  
    void Reset();  
    bool MoveNext();  
    object Current {get;}
```

Class ArrayList : IEnumerable

```
{  
    IEnumerator Getenumerator()  
    {  
        return new Enumerator(this);  
    }  
}
```

Class A

```
{  
    SVM()  
}
```

ArrayList Ar = new ArrayList();

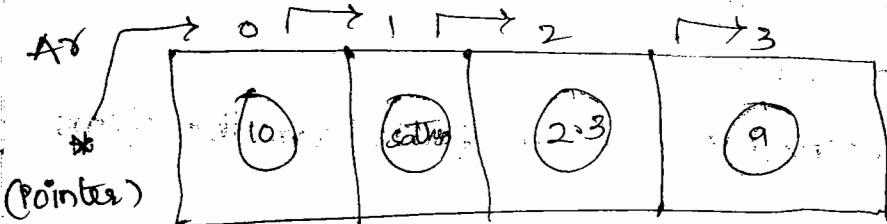
- Ar.Add(10);

- Ar.Add("sathya");

- Ar.Add(2.3);

- Ar.Add('a');

IEnumerator ie = Ar.Getenumerator();



while (ie.MoveNext())

```
{
```

C.WL(ie.Current);

```
}
```

bool MoveNext():- This method is used to move the pointer to the next position.

Object Current:- This property is used to get the current position object

Void Reset():- This method is used to reset the pointer to the initial position.

~~String~~
~~ArrayList~~
~~Verlocution~~
using system;

using System.Collections;

Class Employee : Object

{

int eno;

int ename;

Public Employee(int eno, string ename)

{

this.eno = eno;

this.ename = ename;

}

Public override string ToString()

{

return eno + ename; }

}

Class A

{

Static void Main()

{

ArrayList Ar = new ArrayList();

Ar.Add(new Employee(101, "Anil"));

Ar.Add(new Employee(102, "sunil"));

Ar.Add(new Employee(103, "Ajay"));

IEnumerator ie = Ar.GetEnumerator();

While (ie.MoveNext())

{

Console.WriteLine(ie.Current.ToString()); }

}

}

public virtual string
ToString()

class Object

{

ToString()

GetType()

GetHashCode()

Equals()

};

ToString() will
return current
classname

Eg:- Using System;

Using System.Collections;

Class Employee:object

{

int eno; String ename;

Public Employee (int eno, String ename)

{

this. eno = eno

this. ename = ename;

public int Eno {get {return eno;}}

public String ename {get {return ename;}}

}

Class A

{ static void Main()

{ ArrayList Ar = new ArrayList();

Ar.Add (new Employee(101, "Anil"));

Ar.Add (new Employee(102, "Sunil"));

Ar.Add (new Employee(103, "Ajay"));

IEnumerator ie = Ar.GetEnumerator();

while (ie.MoveNext())

{ Employee el = (Employee) ie.Current;

CWL ("Eno is " + el. Eno);

CWL ("Ename is " + el. Ename);

} }

Forec

collec

Synt

Eg:-

using

Clas

{

SUM

{ Ar

Ar.

Ar.

Ar.

CWL

forea

{ C-1

CWL

for e

forea

{ C-1

CWL

Ar.

fore

{ C-n

8th June 2015

Foreach loop :- foreach loop was introduced for collections.

Syntax :- foreach (var item in collection),
 {
~~foreach~~ ^{variable name} C.WL(item);
 }

~~eg~~ Using system;

using system.collections;

Class A

{

SVM()

{ ArrayList Ar = new ArrayList();

Ar.Add(10);

Ar.Add(20);

Ar.Add(30);

Ar.Add(40);

C.WL("After Add()");

foreach (var item in Ar)

{ C.WL(item); }

C.WL("After Remove(20)");

~~for~~ Ar.Remove(20);

foreach (var item in Ar)

{ C.WL(item); }

C.WL("After insert(170)");

Ar.Insert(170);

foreach (var item in Ar)

{ C.WL(item); }

C.WL("After RemoveAt(2)");

Ar.RemoveAt(2);

foreach (var item in Ar)

{ C.WL(item); }

C.WL("After Reverse()");

Ar.Reverse();

foreach (var item in Ar)

{ C.WL(item); }

C.WL("After Sort()");

Ar.Sort();

foreach (var item in Ar)

{ C.WL(item); }

C.WL("Clear()");

Ar.Clear();

foreach (var item in Ar)

{ C.WL(item); }

Output :- Add(); Remove(20) Insert(170)

0 10	0 10	0 10
1 20	1 30	1 70
2 30	2 40	2 30
3 40		3 40

RemoveAt(2)	Reverse()	Sort()	Clear()
0 10	0 40	0 10	:
1 70	1 70	1 40	:
2 40	2 10	2 70	

Stack :- It follows LIFO Process

Last in First Out Process.

Some of the Predefined methods :-

1. push (object value) :- This method is used to push the items in the stack.
2. pop() :- This method is used to remove the items from the stack.
3. clear() :- This method is used to clear the stack.

Eg:- Using System;

Using System.Collections;

Class A

```
{ SVM()
```

```
{
```

```
Stack S = new Stack();
```

```
S.Push(10);
```

```
S.Push(20);
```

```
S.Push(30);
```

```
S.Push(40);
```

```
S.Pop();
```

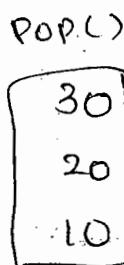
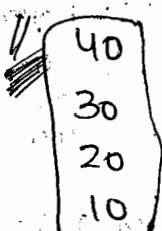
```
foreach (var item in S)
```

```
{
```

```
CW2(item); }
```

```
}
```

Output :-



Ques

7

Ans

1. E

Add

2. D

the

3. C

from

Eg :-

1

Out

Queue:- Queue follows fifo process

First in first out.

Methods:-

1. Enqueue (Object Value) :- This method is used to Add the items in the Queue.
2. Dequeue() :- This method is used to remove the items from the Queue.
3. Clear() :- This method is used to clear the items from the Queue.

eg:- Using System;

Using System.Collections;

Class A

```
{  
    SVM C)
```

```
{
```

```
Queue q = new Queue();
```

```
q.Enqueue(10);
```

```
q.Enqueue(20);
```

```
q.Enqueue(30);
```

```
q.Enqueue(40);
```

```
q.Dequeue();
```

```
foreach (var item in q)
```

```
{
```

```
    Console.WriteLine(item);
```

```
} } }
```

Output :-

→ 20

→ 30

→ 40

HashTable (Object Key, Object Value) :- HashTable will arrange the data in the form of key and value pair format.

```
public struct DictionaryEntry  
{  
    public object Key { get; set; }  
    public object Value { get; set; }  
}
```

All the items in HashTable are of DictionaryEntry type.

Eg:- Using System;

```
using System.Collections;  
class A  
{  
    static void Main()  
    {  
        Hashtable ht = new Hashtable();  
        ht.Add(1, "C");  
        ht.Add(2, "C++");  
        ht.Add(3, "Java");  
        ht.Add(4, "C# .Net");  
    }  
}
```

Output:-

random order

foreach (DictionaryEntry item in ht)

```
{  
    Console.WriteLine("Key is {0} and Value is {1}",  
        item.Key, item.Value);  
}
```

Sorted list :- It is used to arrange the data in ascending order based on key.

Eg:- Using System;

```
using System.Collections;
```

```
class A  
{  
    static void Main()
```

)
s in
re
Entry

```
{  
    SortedList st = new SortedList();  
  
    st.Add(5, "c");  
    st.Add(2, "C++");  
    st.Add(4, "java");  
    st.Add(3, "C#.net");  
    st.Add(1, "sql");  
  
    foreach (DictionaryEntry item in st)  
    {  
        Console.WriteLine("Key is {0} and Value is {1}, item.Key,  
                         item.Value);  
    } } }
```

Output:- Key is 1 and value is sql

2
3
4
5

order

5th Aug 2015

WAP to check whether given string is palindrome or not.

String $\sigma = \text{null};$

CWL ("Enter a string");

String $s = C.RL();$

Char [] Ar = s. TocharArray ();

int length = Ar.length;

for (int i = length - 1; i >= 0; i--)

{

$\sigma = \sigma + Ar[i];$

}

CWL (σ);

if ($\sigma == s$)

{

CWL ("Palindrome");

}

else

{CWL ("Not a palindrome");}

}

WAP to count no of 'a's in the given string

SUM

{

CWL ("Enter a string");

String $s = C.RL();$

Char [] ch = s. TocharArray ();

int count = 0;

for (int i = 0; i <= ch.length - 1; i++)

{

if (ch[i] == 'a')

or not.

{

Count ++;

}

C·WL("no of a's are " + count); } }

WAP to reverse.

Split('char ch') :- This method is used to split the given string based on a character.

e.g:- SVM()

{

String s = "rama is boy";

C·WL ("Enter a word");

String [] Ar = s. Split(' ');

for (int i = Ar.Length - 1; i >= 0; i--)

{

C·W (Ar[i] + " ");

g

C·WL();

there, is, a, cat (Replace , with " ")

WAP to replace , with & " "

SVM()

{

String s = "there, is,a,cat";

s = s. Replace (',', ' ');

C·WL(s);

}

Replace ('oldchar', 'newchar') Syntax for Replace

WAP to display Sathya in the given String.

Svm()

{

String s = "Welcome to Sathya Technologies";

s = s.substring(11);

CWL(s);

~~String~~ s = "Welcome to Sathya Technologies";

s = s.substring(11, 6);

CWL(s);

}

}

Syntax Substring (int startIndex, int length);

Difference between structure and class?

Structure

class

- | | |
|---|---|
| 1. Structure is a userdefined value type datatype | 1. class is a user-defined reference type datatype. |
| 2. Structure must be declared with struct keyword | 2. class must be declared with class keyword. |
| 3. Object is created on stack | 3. Object is created on heap. |
| 4. new keyword is not mandatory to create object | 4. new keyword is mandatory |
| 5. does not support Inheritance | 5. Supports Inheritance. |
| 6. It does not support default constructor but it will supports Parameterised constructor | 6. Class supports both the constructors. |
| 7. structure variable cannot be initialized directly | 7. Class variable can be initialized directly |

13th Aug 2015

Generics:-

- * Generics are called as General datatypes.
- * Generics are used to avoid Function overloading.
- * Generics are used to avoid unnecessary typecasting like Boxing and UnBoxing.
- * Generics are declared with generic notation.
- * we can apply generics for classes, methods, properties, Constructors.
- * In order to work with generics, Microsoft has given predefined namespace
using System.Collections.Generic;
- * Generics are declared by using placeholder and type parameter.
- * Placeholder is used to declare the datatype within angular braces.
- * Type parameter is the parameter that we pass for the generic method.
- * The generic data in the generics is not object type, it is of specific type depending on the data in database. so there is no need of typecasting.

Syntax:- Class classname

{

 void methodname <placeholder>(type parameter)

{

}

}

Eg: Using System;

Using System.Collections.Generic;

Class Program

{

Static void Display <TP> (TP x)

{

Console.WriteLine(x);

Static void Main()

{

Display <int>(10);

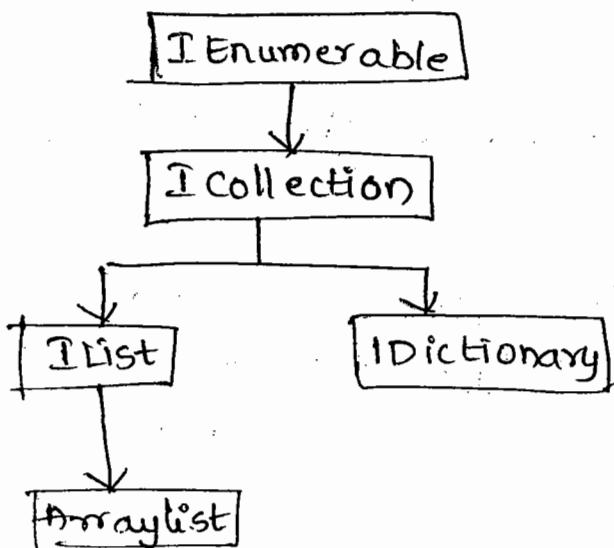
Display <string>("Sathya");

Display <double>(2.3);

Display <char>('a');

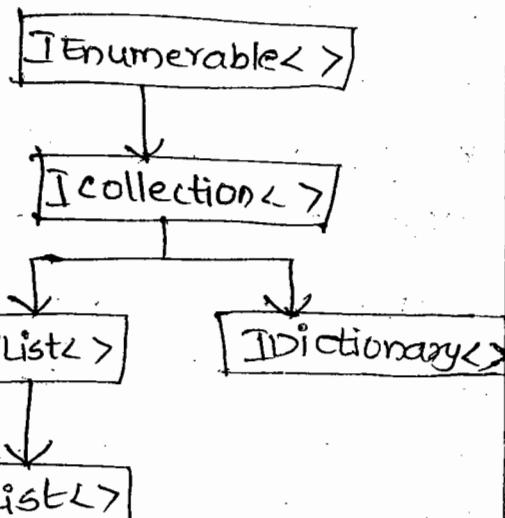
Display <float>(3.4f);

}



Collection type

allows object type of data



Generic type

allows specific type of data

Eg:

C
C
S
LIS

LIS

IE
h

emk

Eg: Using System;

Using System.Collections.Generic;

Class Employee

{

int eno;

String ename;

Public Employee (int eno, String ename)

{

this.eno = eno;

this.ename = ename;

}

Public int Eno

{

get {return eno;}

}

Public int Ename

{

get {return ename;}

}

Class program

{

SVM()

{

List<Employee> emps = new List<Employee>();

emps.Add(new Employee(101, "Anil"));

emps.Add(new Employee(102, "Sunil"));

emps.Add(new Employee(103, "Ajay"));

IEnumerator<Employee> ie = emps.GetEnumerator();

while (ie.MoveNext())

{

Employee el = ie.Current;

CWL(el.eno);

CWL(el.ename); } }

of data

onay <>

Windows Forms Application

[14th Aug 2015]

- * Console Apps are also called as CUI applications
- * (Character User Interface Applications)
- * Console applications are not userfriendly.
- * Windows Forms Applications are called as GUI applications.
- * (Graphical User Interface Applications)
- * Windows Forms Applications are userfriendly.

	WebForms	Windows Forms
1. Design Window	Webform1.aspx [Design]	Form1.cs [Design]
2. Source window	Webform1.aspx [Source]	Form1.Designer.cs
3. Designing code	HTML, ASP.NET	C# .NET
4. Business logic	Webform1.aspx.cs	Form1.cs
5. Sol ⁿ Explorer	✓	✓
6. Properties Window	✓	✓
7. Events Window	✓	✓
8. Toolbox	✓	✓
9. Server Explorer	✓	✓
10. Errorlist	✓	✓

Validations in Windows Forms App:-

[15th Aug 2015]

a - z	97 - 122
A - Z	65 - 90
0 - 9	48 - 57
.	46
@	64
-	95
backspace	8

Enter Username

Enter Phno

Button1

Requirement :-

1. Username must not accept only lowercase, . , @ and -
Characters
2. Phno must accept only numbers

Whenever user press any key in the keyboard
then KeyPress event will fire

Select TextBox1 → Properties → events → DC on
KeyPress event and write the code

Private void TextBox1_KeyPress (Object sender, KeyPress
EventArgs e)

}

KeyPress → Event

Event will call delegate

KeyPressEventHandler → Delegate

Delegate will call method

TextBox1_KeyPress → method.

Object Sender :- Sender is a variable of object type.

KeyPressEventArgs e :- e is the reference variable

of KeyPress EventArgs class.

Public Class KeyPressEventArgs : EventArgs

{

 Public bool Handled { get; set; }

 Public char Keychar { get; set; }

}

* Handled Property is a boolean property which will return either true or false.

e. Handled = True :- When this property is set to true then the typed character will disappear within the textbox when false the typed character will appear by default it is false.

Keychar Property :- This property is used to get the ASCII value of the typed character within the TextBox.

Struct char

```
{  
    public static bool IsDigit (char c)  
    {  
        if (c >= 48 && c <= 57)  
        {  
            return true;  
        }  
        else  
        {  
            return false;  
        }  
    }
```

```
public static bool ISLower (char c) { }
```

```
public static bool ISUpper (char c) { }
```

```
public static bool ISCharacter or Number
```

Select textBox1 → Properties → Events → Click on KeyPress event and write the code.

b will
take
e
or by
de
box.

```
private void textBox1_KeyPress( )  
{  
    if (char.IsLower(e.KeyChar) == false &&  
        Convert.ToInt32(e.KeyChar) != 8 &&  
        Convert.ToInt32(e.KeyChar) != 46 &&  
        Convert.ToInt32(e.KeyChar) != 64 &&  
        Convert.ToInt32(e.KeyChar) != 95)  
    {  
        MessageBox.Show("invalid username");  
        e.Handled = true;  
    }  
}
```

Select textBox2 → properties → Events → On
KeyPress event and write the code.

```
private void textBox2_KeyPress( )  
{  
    if (char.IsDigit(e.KeyChar) == false &&  
        Convert.ToInt32(e.KeyChar) != 8)  
    {  
        MessageBox.Show("Please enter only numbers");  
        e.Handled = true;  
    }  
}
```

50

Enter Username

Enter Password

Confirm Password

Enter Age

Select TextBox1.Text → Properties → Events → Leave Event.

Leave Event :- This Event will fire whenever the control leaves the textbox.

Focus() :- this method is used to focus the cursor within the textbox control.

```
private void textBox1_Leave()
{
    if (textBox1.Text == " ")
    {
        MessageBox.Show("Username must not be empty");
        textBox1.Focus();
    }
}
```

```
private void textBox2_Leave()
{
    if (textBox1.Text == textBox2.Text)
    {
}
```

```
    MessageBox.Show("Username and password must not be
                     same");
    textBox2.Focus();
}
```

Properties
→ PasswordChar *

```
private void textBox3_Leave()
{
    if (textBox2.Text != textBox3.Text)
    {
        MessageBox.Show("Password mismatch");
        textBox3.Focus();
    }
}

private void textBox4_Leave()
{
    if (textBox4.Text == " ")
    {
        MessageBox.Show("Age must not be empty");
        textBox4.Focus();
    }

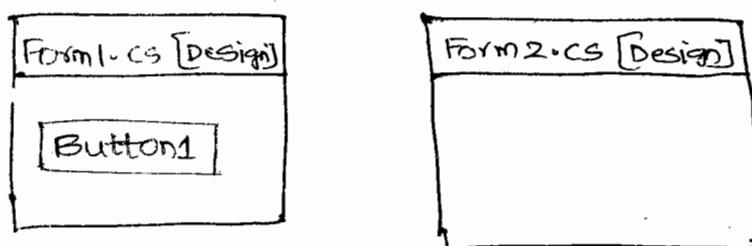
    int age = int.Parse(textBox4.Text);
    if (age >= 18 && age <= 25)
    {
    }
    else
    {
        MessageBox.Show("Age must be between 18 and 25");
        textBox4.Focus();
    }
}
```

it be
→

17th Aug 2015

Q) How to Redirect the user request between multiple forms?

By creating an object for the destination Form.



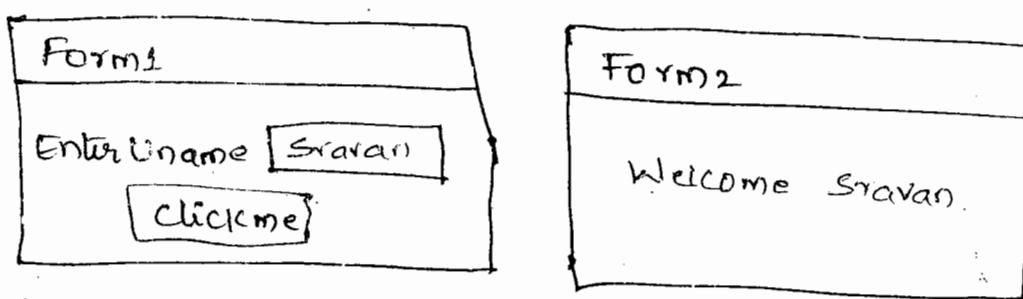
Button1 - click()

{

```
Form2 f = new Form2();
f.show();
}
```

Q) How to maintain the values b/w multiple forms in windows Forms Application?

1. Public Static Variables
2. Public Static properties
3. Parameterised constructor.



Code for Form1.cs :-

```
Public Partial class Form1 : form
{
    Public static string uname;
    Public Form1()
    {
        Initialize Component();
    }
}
```

```
Private void Button1_Click()
{
    uname = TextBox1.Text;
    Form2 f = new Form2();
    f.Show();
}
```

Code for form2.cs :-

```
public partial class Form2 : Form
{
    public Form2()
    {
        InitializeComponent();
    }

    private void Form_Load()
    {
        Label1.Text = "Welcome" + Form1.uname;
    }
}
```

Code for Program.cs :-

```
static class Program
{
    [STAThread]
    static void Main()
    {
        Application.Run(new Form1());
    }
}
```

Steps :-

1. Program execution starts from Main()
2. new Form1() object is created for Form1
3. Form1() Constructor is called and InitializeComponent method will get executed and make Form1 Ready

4. on
5. c
6. f

Call
f. F
ia
2. M
codi
Pub
{
Pub
Priv
{
un
fc
=

Form
==
priv
{
labe
}
3. Ma
Form
priv
{
}

4. User will enter Username in textbox and click on Button then Button1_Click() will fire.

5. Store the Value in static Variable

6. Form2 f = new Form2();

Object is created for Form2 and Form2() is called and make Form2 ready.

7. Form2_Load() will call and display Username in label.

2. Maintaining the values by using public static property:-

Code for form1.cs :-

```
public partial class Form1 : Form  
{
```

```
    public static string Uname { set; get; }
```

```
    private void button1_Click()  
{
```

```
        Uname = TB1.Text;
```

```
        Form2 f = new Form2();
```

```
        f.Show();
```

```
} }
```

Form2.cs :-

```
private void Form2_Load()  
{
```

```
    Label1.Text = "Welcome" + Form1.Uname;  
}
```

3. Maintaining the values by using parameterised constructor:-

Form1_Load :-

```
private void button1_Click()  
{
```

```
    Form2 f = new Form2(textBox1.Text);  
    f.Show();
```

Code for form2 :-

```
Public Form2 (String uname)
{
    Initialize Component();
    Label1.Text = "welcome" + Uname;
}
```

Progress Bar Control :-

It is used to show the progress of the output.

Properties :-

Value :- This property is used to set the current position of the progressBar control.

Timer Control :- It is used to halt the output within a particular time period.

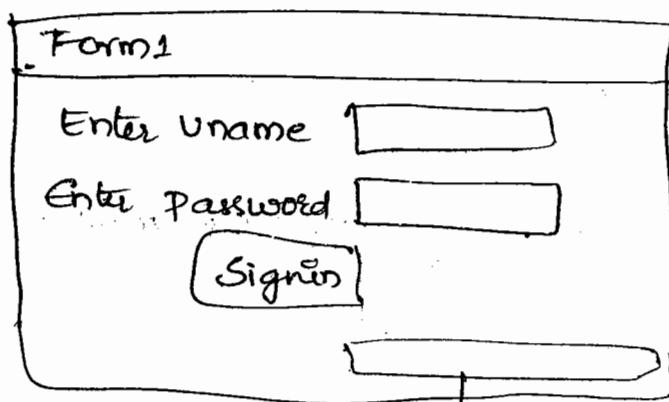
Properties :-

1. Enabled = True/False :- If true, timer will work.

2. Interval :- This property is used to set the time interval in milliseconds.

Events :-

1. Tick Event :- This event will fire when timer enabled = True.



Timer Control

ProgressBar

DC on button & write the code

```
Private Void button1_Click()
{
    if (tb1.Text == "Admin" & tb2.Text == "Admin")
    {
        timer1.Enabled = true;
    }
    else
    {
        MessageBox.Show("invalid user");
    }
}
```

Within Select timer → properties → DC on Tick event & write code.

```
Private Void timer1_Tick()
{
    progressBar1.Value = progressBar1.Value + 5;
    if (progressBar1.Value >= 100)
    {
        timer1.Enabled = false;
        Success s = new Success();
        s.show();
    }
}
```

18th Aug 2015

Display the data within the datagridview control:

Step-1: Drag and Drop Datagridview Control from Toolbox

Step-2:- goto → SQL explorer → App.config and declare
the connection string.

<configuration>

<connection strings>

```
<add name = "constr" ConnectionString = "User id = sa;  
password = abc ; database = sathya ; datasource = Kannababu" />
```

</connection strings>

Step 3:- goto → SQL explorer → References → Add Reference
→ Select System.Configuration → OK.

Step 4:- using System.Data.SqlClient;
using System.Configuration;
private void FillData()

```
{  
    SqlConnection con = new SqlConnection(ConfigurationManager.  
        ConnectionStrings["constr"].ToString());
```

```
    SqlDataAdapter da = new SqlDataAdapter("select * from emp",  
                                         con);  
    DataSet ds = new DataSet();  
    da.Fill(ds, "emp");
```

```
    dataGridView1.DataSource = ds.Tables["emp"];
```

```
}
```

```
private void Form1_Load()
```

```
{  
    FillData()  
}
```

5

Step-5:- Add Delete Button and Edit Button within DataGridView Control.

Toolbox

Select DataGridView → ellipse button → Edit columns → Add Name = Delete.

large

Type = DataGridViewButtonColumn

Header text = Delete

Select delete → properties

SA;

mababu")>

Text = Delete

use column text for button = true

Similarly add edit button

reference Step-6:- Whenever user clicks on any button control within the DataGridView then cell content click event will fire

0	1	2	3	4	column index
delete	edit	Eno	Ename	Salary	
delete	(edit)	101	Anil	23000	

* Whenever user clicks on Delete button.

m emp) 1. Count the no. of rows that are available within the gridview control

* Rows property will count the no of rows.

2. Catch the index value of the row where user clicks on delete button.

* e. RowIndex is used to catch the RowIndex.

3. catch the cell based on which we want to perform the operation. * Cells[2]

4. catch the value i.e available within the cell value property is used to catch the value within the cell.

Step 7:- Select datagridview control → Properties →
Events → double click on cellcontent click event
and write the code.

Sql

```
private void DataGridView1_CellContentClick() {
    if (e.ColumnIndex == 0) {
        int empno = Convert.ToInt32(DataGridView1.
            Rows[e.RowIndex].Cells[2].Value);
        SqlConnection con = new SqlConnection();
        con.Open();
        String query = "delete from emp where eno=" + empno;
        SqlCommand cmd = new SqlCommand(query, con);
        cmd.ExecuteNonQuery();
        con.Close();
        FillData();
    }
    else {
        int empno = Convert.ToInt32(DataGridView1.
            Rows[e.RowIndex].Cells[2].Value);
        String emphname = DataGridView1.Rows[e.RowIndex].
            Cells[3].Value.ToString();
        double empsal = Convert.ToInt32(DataGridView1.
            Rows[e.RowIndex].Cells[4].Value);
        SqlConnection con = new SqlConnection();
        con.Open();
```


Collection Controls

19th Aug 2015

1. Combobox control

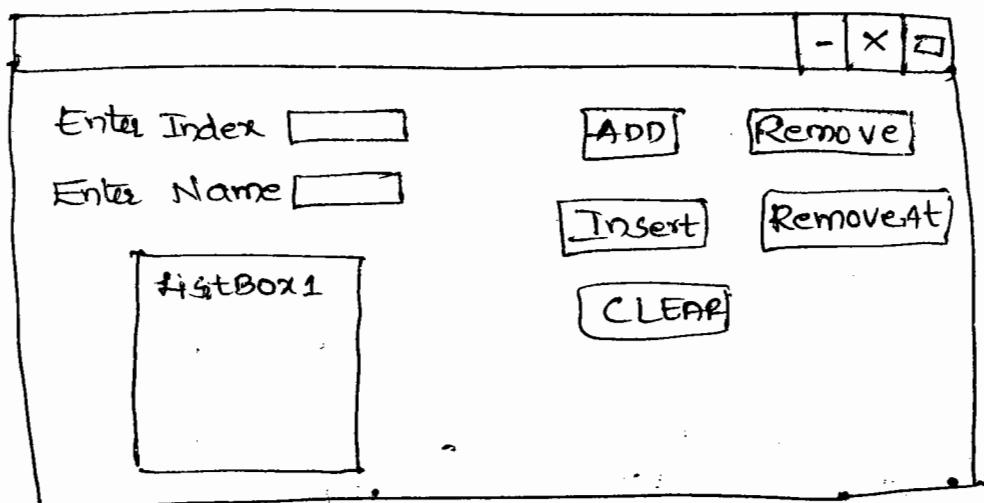
2. Listbox control.

Listbox Control:-

It is used to select one item or more than one item or Group of item.

Combobox Control:-

It is the combination of textbox and dropdownlist.



Private void button1_Click()

{

listBox1.Items.Add(textBox2.Text);

}

Private void button2_Click()

{

listBox1.Items.Remove(textBox2.Text);

}

Private void button3_Click()

{ int index = int.Parse(textBox1.Text); }

listBox1.Items.Insert(index, textBox2.Text);

}

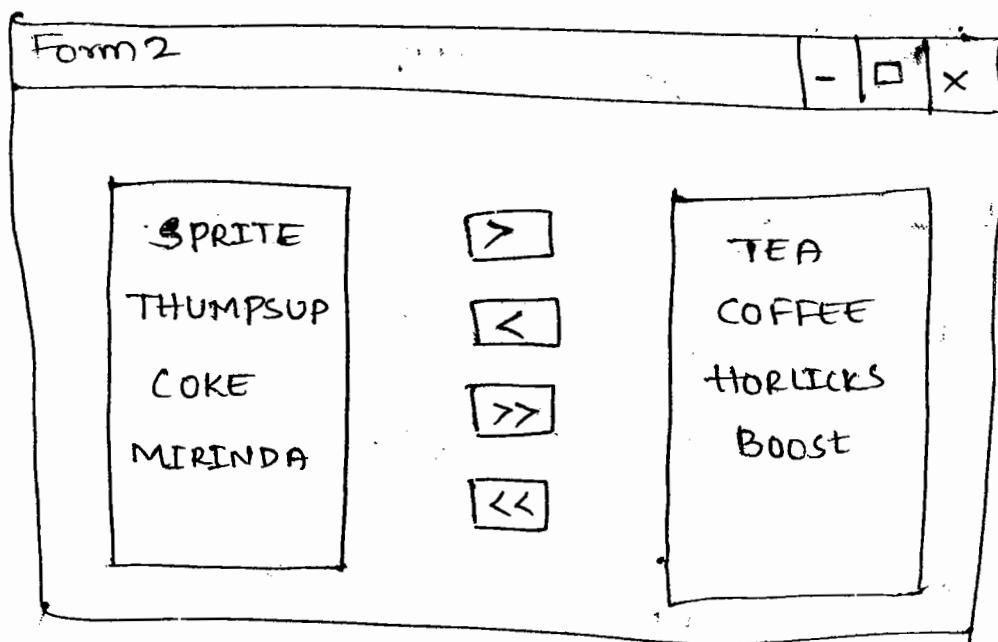
```

private void button4_Click()
{
    int index = int.Parse(textBox1.Text);
    listBox1.Items.RemoveAt(index);
}

private void button5_Click()
{
    listBox1.Items.Clear();
}

private void button6_Click()
{
    this.Close();
}

```



```

private void button1_Click()
{
    listBox2.Items.Add(listBox1.SelectedItem.ToString());
    listBox1.Items.Remove(listBox1.SelectedItem.ToString());
}

```

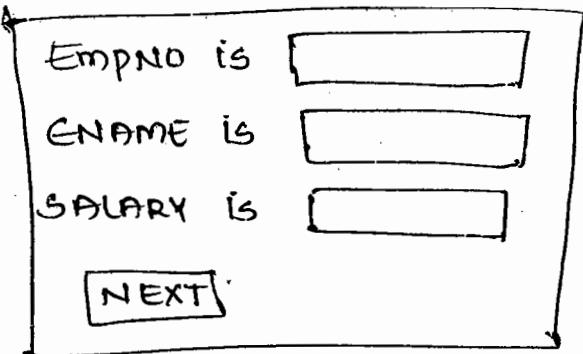
```
Private void button2_Click(Object sender, EventArgs e)
{
    listBox1.Items.Add(listBox2.SelectedItem.ToString());
    listBox2.Items.Remove(listBox2.SelectedItem.ToString());
}

private void button3_Click()
{
    for (int i = 0; i < listBox1.Items.Count - 1; i++)
    {
        listBox2.Items.Add(listBox1.Items[i].ToString());
    }
    listBox1.Items.Clear();
}

private void button4_Click()
{
    foreach (var item in listBox2.Items)
    {
        listBox1.Items.Add(item);
    }
    listBox2.Items.Clear();
}

(or)

private void button4_Click()
{
    for (int i = 0; i < listBox2.Items.Count - 1; i++)
    {
        listBox1.Items.Add(item);
    }
    listBox2.Items.Clear();
}
```



```
using System.Data.SqlClient;
using System.Configuration;
public partial class Form3 : Form
{
    SqlConnection Con;
    SqlDataReader dr;
    SqlCommand Cmd;
    private void Form3_Load()
    {
        SqlConnection Con = new SqlConnection(
            Manager.Connection
        );
        Con.Open();
        Cmd = new SqlCommand("Select *");
        dr = Cmd.ExecuteReader();
        if (dr.HasRows)
        {
            if (dr.Read())
            {
                textBox1.Text = dr[0].ToString();
                textBox2.Text = dr[1].ToString();
                textBox3.Text = dr[2].ToString();
            }
        }
    }
}
```

```
Private void button1_Click()
{
    if (dr.Read())
    {
        TextBox1.Text = dr[0].ToString();
        TextBox2.Text = dr[1].ToString();
        TextBox3.Text = dr[2].ToString();
    }
    else
    {
        MessageBox.Show("no more records");
    }
}
```

tion

.ToString());

);

Advanced C++

Object Initializer :- It is used

16th Aug 2015

to initialize the values for the automatic properties at the time of creating object.

e.g:-

```
using System;
class emp
{
    public int Eno {get; set;}
    public string Ename {get; set;}
}
```

class Program

{

Static void Main()

```
    Emp e1 = new Emp() { Eno=101, Ename="mohan"};
```

```
    C.WL ("Eno is" + e1.eno);
```

```
    C.WL ("Ename is" + e1.Ename);
```

}

}

5

Collection Initializers:-

It is used to initialize the values for collection.

* Collection Initializer is the combination of object Initializer.

e.g:- Using System;

```
using System.Collections.Generic;
```

```
class Employee
```

```
{
```

```
    public int Eno {get; set;}
```

```
    public string Ename {get; set;}
```

```
}
```

Class Program

```
{
```

```
static void Main()
```

```
{
```

```
list<Employee> emps = new List<Employee>()
```

```
{
```

```
    new Employee {Eno=101, Ename="Anil"},
```

```
    new Employee {Eno=102, Ename="sunil"},
```

```
    new Employee {Eno=103, Ename = "Ajay"},
```

```
};
```

foreach (Employee item in emps)

```
{
```

```
    C.WL ("Eno is" + item.Eno);
```

```
    C.WL ("Ename is" + item.Ename);
```

```
} }
```

Extension methods :-

Extension methods are used to add a new method to the existing class without modifying the original class (ie without Inheritance)

Rules to declare the extension methods:-

- * An extension method must be declared in static class.
 - * Extension method must be static method.
 - * While declaring extension method this keyword is applied for the first parameter.

Syntax:-

Static class classname

९

public static returntype methodname (this

7

3

2

Eg:- Using system;

class A

8

Public void Show()

9

CWL ("I am Show"); }

```
Public void Display()
```

{ CWL("I am Display"); }

Static class xxx

{

 Public static void NewMethod (this A obj)

{

 Console.WriteLine ("I am extension method");

} }

Class program

{

 Static void Main ()

{

 A a1 = new A ()

 a1.Show ();

 a1.Display ();

 a1.NewMethod ();

} }

Note :- Sealed class will not participate in Inheritance
If we want to add some extra methods to the sealed class then we can use extension methods.

Anonymous Types :-

Anonymous Types are implicitly typed local variables.

* Anonymous Types allow us to create a new type without defining them.

* Anonymous Type must be declared with Var keyword.

* Always Var keyword must declare as local variable

* Generally we will use Anonymous types in foreach loop and link queries.

Eg: Class A

```

{
Static void Main()
{
Var a=10;
Var b="sathya";
Var Ar=new int[3] {10,20,30};
CWL(a);
CWL(b);
CWL foreach (Var item in Ar)
{
    CWL(item);
}
}

```

Eg: Using System;

Class A

```

{
SVM()
{
}

```

Var Obj = new {Eno=101, Ename="Anil"};

CWL(Obj. Eno);
CWL(Obj. Ename);

}

Anonymous class:-

Anonymous class is a class that has no name

* We can create an object for the anonymous class using new keyword and a pair of braces defining the fields and values that we want the class to contain

Refer above example.

An
 blo
 for
 * A
 code
 a a
 bloc
 Bene
 * Ti
 * Ti
 * Ti

with
 usi
 Publi
 class
 {
 sta
 {
 CWI
 }
 Sta
 {
 MyD
 ob
 }

Anonymous methods :- Anonymous method is a block of code that is used as the parameter for the Delegate.

* Anonymous method allows coder to pass block of code rather than the name of the method. Creating a anonymous method is a way to pass a code block as delegate parameter.

Benefits :-

- * To reduce the amount of code
- * To reduce complexity of code
- * To increase readability of code

Without Anonymous method

using System;

Public delegate void MyDelegate();

class A

{

 Static void Show()

{

 Console.WriteLine("I am Show");

}

 Static void Main()

{

 MyDelegate obj = new

 MyDelegate>Show());

 obj();

}

With Anonymous method.

using System;

Public delegate void Mydelegate();

class Program

{

 Static void Main()

{

 MyDelegate obj = delegate()

{

 Console.WriteLine("I am Show");

};

 obj += delegate()

{

 Console.WriteLine("I am Display");

};

 obj();

}

Anonymous method with return type

Using System;

Public delegate void MyDelegate (int x, int y);

Class program

{

SVM()

{

MyDelegate obj = delegate (int x, int y)

{

cWL (x+y);

}

obj (10, 20);

}

Anonymous method with Events :-

Using System;

Public delegate void MyDelegate (int x, int y);

Class program

{

Public static event MyDelegate MyEvent;

SVM()

{

MyEvent += delegate (int x, int y)

{

cWL (x+y);

}

MyEvent (10, 20);

}

}

dar

=

Con

Syn

I

J

eg:-

1.

2. (

3.

4.)

5.

6.

7.

eg:-

Put

Pub

Publ

eg:-

Put

pu

Put

Cle

{

Lambda Expression :- Lambda Expression at Compile time converts to anonymous method.

Syntax :-

I/p Parameters \Rightarrow method body or expression
 \downarrow
Lambda Operator.

I/p Parameters goes to method body
 \Rightarrow

eg:-

1. $(x, y) \Rightarrow x = y;$
2. $(x, y) \Rightarrow \text{return } x + y;$
3. $() \Rightarrow \text{CWL}(\text{"Hello Lambda"});$
4. $(\text{int } x, \text{int } y) \Rightarrow \text{return } x > y;$
5. $z \Rightarrow \text{return } z + 9;$
6. $x \Rightarrow x = 9; \text{CWL}(x);$
7. $() \Rightarrow \text{return } \text{"hello"};$

eg:- Using System;

Public static event MyDelegate; ~~MyDelegate~~
Public static event MyDelegate1 (int x);
Public static event i;

eg:- Using System;

Public delegate void MyDelegate();
Public delegate void MyDelegate1 (int x);
Public delegate int MyDelegate2 (int x, int y);

Class Program

{

Public static event MyDelegate MyEvent;
Public static event MyDelegate1 MyEvent1;
Public static event MyDelegate2 MyEvent2;
SVM()
{

MyEvent += () => CWL("Hello I am Lambda");
MyEvent();
MyEvent1 += x => CWL(x);
MyEvent(10);
MyEvent2 += (x, y) => x + y;
int z = MyEvent2(10, 20);
CWL(z);

}

}

Q) +
for
By

Q) +
in e
1. P
2. P
3. T

F
E

Cod.
Pub
{
Put
Pu
{
I

Form1.Designer.cs

partial class Form1 → Derived class

```
{
private void InitializeComponent()
{
    this.Name = "myform";
    this.Text = "myform1";
    this.Width = 300;
    this.Height = 200;
    this.ForeColor = "red";
    this.Font = "Times New Roman"
}
```

~~this(Button).click += new EventHandler
(Button1_Click);~~

program.cs

static class program

{

static void Main()

{

Application.Run(new Form1());

}

Form1.cs

partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

protected void Button1_Click

(object sender, EventArgs e)

{

MessageBox.Show("I am
Button");

}

class Form → Base
class

{

Name;

Height;

Text;

ForeColor;

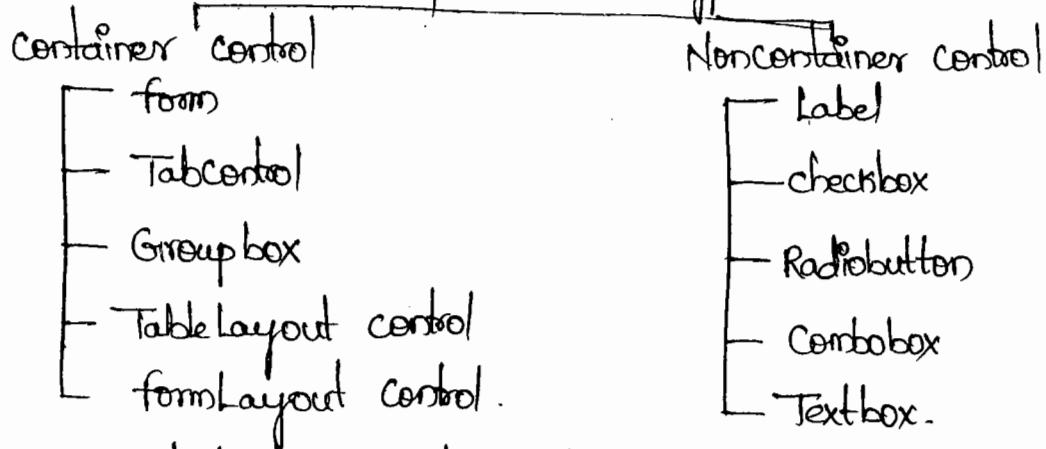
Font;

Width;

Controls:-

Control:- Control is a predefined class which is used for designing the windows form.

Controls are divided into two types.



Container control:- The controls which can hold other controls in them are called as container controls.

Ex: form, Groupbox etc.

NonContainer control:- The controls which can't hold other controls in them are called as Noncontainer controls.

Ex: Label, Textbox etc.

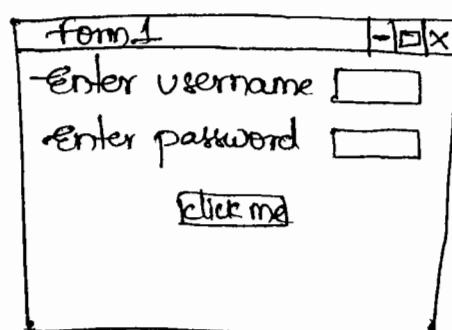
Message Box·show():- Message Box is a predefined class.

show() is a static method which will be called with class name.

MessageBox·show() is used to display a friendly message to the user with OK button.

Ex: Double click on button and click me
protected void button1_Click()

```
{  
    message Box·show ("welcome"+ Tb.Text);  
}
```



Q) How to navigate the user request from one form to another form?

If we want to navigate the user request b/w multiple forms we have to create an object for the destination form.

```
class form1: form
{
    public form1()
    {
        initialize component();
    }

    private void button1_Click()
    {
        Form2 f = new Form2();
        f.show();
    }
}

class form2: form
{
    public form2()
    {
        initialize component();
    }
}

class program
{
    static void Main()
    {
        Application.Run(new form1());
    }
}
```

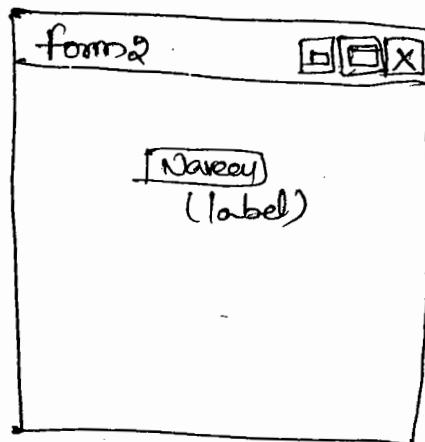
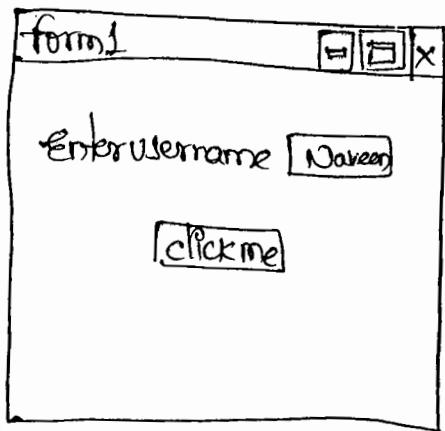
steps for execution:-

1. program execution starts from Main() method.
2. new form1(): form1() constructor is invoked and InitializeComponent() method is called and making the form1 ready and displaying the form1 to user.
3. whenever user clicks on button internally foll. code is generated.
button1.click = new EventHandler(Button1_Click)
↓ ↓ ↓ ↓
objectname event delegatename method name
4. click is an event which call eventhandler delegate which internally calls the Button1_Click() method and the code which we've written inside Button1_Click() will gets executed.

5. In Button1_Click we have created an object for form2 so form2() constructor will be invoked and InitializeComponent() will be called and makes form2 ready to the user.

Maintaining the values b/w multiple forms in windows application:-

1. public static variables
2. properties 3. constructors



Code for the form1.cs:

```
public partial  
class form1 : form  
{  
    public static string name;  
    public form1()  
    {  
        InitializeComponent();  
    }  
    private void Button1_Click()  
    {  
        name = Tb1.Text;  
        form2 f = new form2();  
        f.show();  
    }  
}
```

Code for the program.cs:

```
static class program  
{  
    static void Main()  
    {  
        Application.Run(new form1());  
    }  
}
```

Code for form2.cs:

```
public partial class form2 : form  
{  
    public form2()  
    {  
        InitializeComponent();  
    }  
    private void form1_Load()  
    {  
        Label1.Text = "Welcome" +  
            form1.name;  
    }  
}
```

Maintaining the values b/w multiple forms using properties:-

Code for Form1.cs:

```
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
    }  
  
    public static string Username  
    {  
        set;  
        get;  
    }  
  
    private void Button1_Click()  
    {  
        Username = Tb1.Text;  
        form2 f = new form2();  
        f.Show();  
    }  
}
```

Code for Form2.cs:

```
public partial class Form2 : Form  
{  
    public Form2()  
    {  
        InitializeComponent();  
    }  
  
    private void Form2_Load()  
    {  
        Label1.Text = Form1.Username;  
    }  
}
```

Code for program.cs:

```
static class program  
{  
    static void Main()  
    {  
        Application.Run(new Form1());  
    }  
}
```

Maintaining the values b/w multiple forms using constructors:-

Code for Form1.cs:

```
public partial class Form1 : Form  
{  
    public Form1()  
    {  
        InitializeComponent();  
    }  
    private void Button1_Click()  
    {  
        Form2 f = new Form2(TextBox1.Text);  
        f.Show();  
    }  
}
```

Code for Form2.cs:

```
public partial class Form2 : Form  
{  
    public Form2(string name)  
    {  
        InitializeComponent();  
        Label1.Text = "welcome" + name;  
    }  
    private void Form2_Load()  
    {  
    }
```

Code for program.cs:

```
static class program  
{  
    static void Main()  
    {  
        Application.Run(new Form1());  
    }  
}
```

Form1

Enter username

Select hobbies playing
 singing dancing

Select gender male female

Form2

U.n. is

Hobbies are

Gender is

Code for Form1.cs:

```
public partial class Form1 : Form
{
    public Form1() { static string gender; }

    protected void InitializeComponent()
    {
        public static string Username
        {
            set; get;
        }

        public static string Hobbies
        {
            set; get;
        }

        private void Button1_Click()
        {
    }
```

✓ (Username = TextBox1.Text;)
 if (cb1.checked == true)
 {
 hobbies = cb1.Text;
 }
 if (cb2.checked == true)
 {

```
    hobbies = hobbies + cb2.Text;  

    }  

    if (cb3.checked == true)  

    {  

        hobbies = hobbies + cb3.Text;  

    }  

    if (cb4.checked == true)  

    {  

        hobbies = hobbies + cb4.Text;  

    }  

    form2 f = new form2();
    f.show();
}
```

Code for form2.cs:

```
public partial class Form2 : Form  
{  
    public Form2(string gender)  
    {  
        InitializeComponent();  
        Label1.Text = "Gender is " + gender;  
        Label1.Text = "Username is " + Username;  
        Label2.Text = "Hobbies are " + hobbies;  
    }  
}
```

Code for program.cs:

```
static class program  
{  
    static void Main()  
    {  
        Application.Run(new Form1());  
    }  
}
```

Validations in windows app:-

Validation:- It is a process of restricting the control by accepting a proper i/p from the user.

Ex:

(i) Username must not be empty.

Leave event:- This event belongs to Tb control. This event will fire whenever the control leaves the Tb.

Focus():- This is a method belongs to Tb control, this method

is used to focus the control within the textbox.

goto → form1.cs [Design]

Select Tbl → properties → events →

click on Leave event and write the code.

```
private void textBox1_Leave()
```

```
{  
    //Username must not be empty
```

```
    if (textBox1.Text == "")
```

```
{
```

```
    MessageBox.Show("Username must not be empty");
```

```
    textBox1.Focus();
```

```
}
```

```
}
```

```
// private void textBox2_Leave()
```

```
{
```

```
    //password must not be empty
```

```
    if (textBox2.Text == "")
```

```
{
```

```
    MessageBox.Show("password must not be empty");
```

```
    textBox2.Focus();
```

```
}
```

```
// username and pwd must not be same
```

```
if (textBox1.Text == textBox2.Text)
```

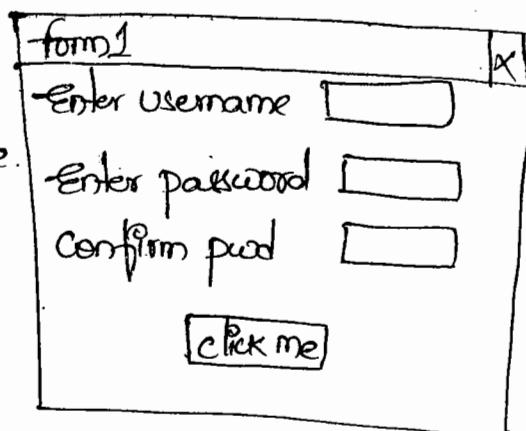
```
{
```

```
    MessageBox.Show("username & pwd must not be same");
```

```
    textBox2.Focus();
```

```
}
```

```
}
```



```

private void textBox3_Leave()
{
    // password mismatch
    if (textBox2.Text != textBox3.Text)
    {
        MessageBox.Show("password mismatch");
        textBox3.Focus();
    }
}

```

Key press event:- This event belongs to Tb control. Keypress event will fire whenever user press any key within the Tb control.

→ `private void TextBox1_KeyPress(object sender, KeyPressEventArgs e)`

```
{
}
```

TextBox1_KeyPress: This method consists of two parameters.
object sender: Sender is a variable of object type telling us the type of control that we are working with.

keypress event args e: e is an object of keypress event args class telling us the type of event that we are working with.

This class consists of two properties

(i) keychar: This property is used to get the ASCII value of the typed character within the textbox control.

ASCII value

@ 64

a-z

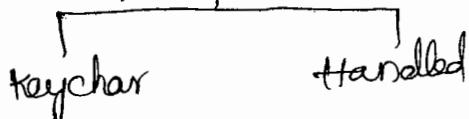
· 46

A-Z

- 95

0-9

< 8



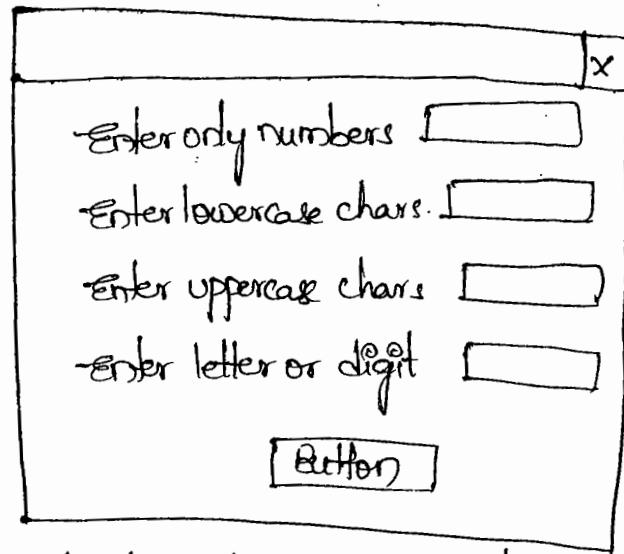
char class: This class provides some set of predefined methods.

IsDigit(): This method will accept the ASCII values b/w 48-57

IsLower(): " " " 97-122

IsUpper(): " " " 65-90

All the above methods are static methods.



Ex: private void TextBox1_KeyPress(Object sender, KeyPressEventArgs e)

{

if (char.IsDigit (e.KeyChar) == false) && convert.ToInt32 (e.KeyChar) != 8

{

MessageBox.Show("enter only numbers");

e.Handled = true;

}

}

private void Tb2_KeyPress()

{

if (char.IsLower (e.KeyChar) == false && convert.ToInt32 (e.KeyChar) != 8)

{

MessageBox.Show("please enter only lowercase characters");

e.Handled = true;

}

}

```
private void Tb3_KeyPress()
{
    if (char.IsUpper(e.KeyChar) == false && Convert.ToInt32(e.KeyChar) != 8)
    {
}
```

```
    MessageBox.Show("please enter only uppercase characters");
    e.Handled = true;
}
```

```
}
```

```
private void Tb4_KeyPress()
{
    if (char.IsLetterOrDigit(e.KeyChar) == false && Convert.ToInt32(e.KeyChar) != 8)
    {
}
```

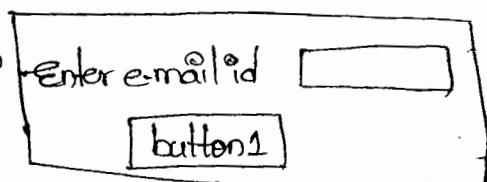
```
    MessageBox.Show("please enter letter or digit");
    e.Handled = true;
}
```

```
}
```

Email id validation:-

goto ~~create~~ new project → windows app

→ design the foll. in form.



```
private void TextBox1_KeyPress()
{
```

```
    if (char.IsLower(e.KeyChar) == false && Convert.ToInt32(e.KeyChar) != 8)
        && Convert.ToInt32(e.KeyChar) != 64 && Convert.ToInt32(e.KeyChar) != 46
        && Convert.ToInt32(e.KeyChar) != 95)
    {
```

```
        MessageBox.Show("Invalid e-mail id");
        e.Handled = true;
}
```

```
}
```

User controls:- The controls that are created by the user depending on the user requirement are called as user controls.

Steps to create user defined event :-

1. Create a delegate.

```
public delegate void myDelegate();
```

2. Create an event.

```
public event myDelegate Register;
```

3. Create a method for delegate.

```
public void Method()
```

```
{
```

```
}
```

4. Call the event

```
eventname,
```

Creating a Register control and consuming it in windows form:-

Step 1: go to → form1.cs [Design] → project → Add user control →

Select user control template Name: Registration control.cs → Add.

Design the control:-

The diagram shows a rectangular window titled "Registration Control". Inside, there are four text input fields arranged vertically. Each field has a label to its left and a corresponding empty text box to its right. At the bottom of the window is a single button labeled "Register".

Enter username	<input type="text"/>
Enter password	<input type="text"/>
Confirm password	<input type="text"/>
Enter address	<input type="text"/>

Register

performing validations for Register Control:-

Select → TB₁ → properties → Events → D·c on Leave event & write the code:

public partial class Registercontrol : user control

```
{  
    public static string uname;  
    public static string pwd;  
    public static string cpwd;  
    public static string Address;
```

// create a delegate

```
public delegate void Mydelegate();
```

// create an event

```
public event Mydelegate Register_click;
```

```
private void TB1_Leave()
```

```
{  
    if (TB1.Text == "")
```

```
{
```

```
    MB.show("Username must not be empty");
```

```
#
```

```
    TB1.focus();
```

```
    #
```

```
    #
```

If for TB₂:

```
private void TB2_Leave()
```

```
{
```

```
if (TB1.Text == TB2.Text)
```

```
{
```

```
    MB.show("Username & pwd must not be same");
```

```

TB2.Focus();
}
}

// for TB3
private void TB3_Leave()
{
if (TB2.Text != TB3.Text)
{
mb.show("password mismatch");
TB3.Focus();
}
}

private void button1_Click()
{
uname = TB1.Text;
pwd = TB2.Text;
cpwd = TB3.Text;
Address = TB4.Text;
// Call the event
Register_Click();
}
}

```

click → Build → Build sol.

goto form1.cs [Design] → drag & drop register control from tool box in [Design] → select Register Control → properties → events → D.C on Register_Control event & write the code.

```

using system.data.sqlclient;
private void registercontrol1_Register_Click()
{
    sqlconnection con = new sqlconnection();
    con.open();
    string query = "insert into register values (" + Registercontrol1.Uname +",
    "+ Registercontrol1.Pwd + ", " + Registercontrol1.Cpwd + ", " + Registercontrol1.Address + ")";
    SqlCommand cmd = new SqlCommand(query, con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Record is inserted");
    con.Close();
}

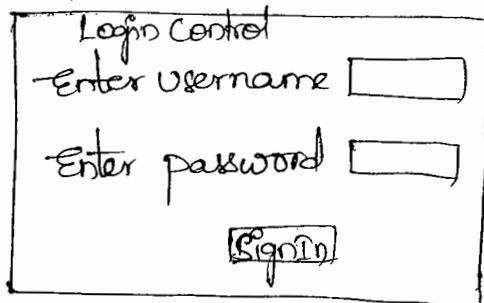
```

Creating a Login user control & consuming in windows forms app:-
 goto → form1.cs [Design] → goto → project → Add user control
 → select template = user control → name = login.cs → add.

* (Select TextBox1 → properties →) x
 goto → login.cs & write the code

```

public partial class login : usercontrol
{
    public static string Uname;
    public static string password;
    // Create a delegate
    public delegate void mydelegate();
    // Create an event
    public event mydelegate login_Click;
  
```



//create a method for delegate x

```
private void TB1_Leave()
{
    if (TB1.Text == "")
    {
        MB.show("username must not be empty");
        TB1.Focus();
    }
}
```

```
private void TB2_Leave()
{
```

```
    if (TB2.Text == "")
    {
        MB.show("password must not be empty");
        TB2.Focus();
    }
}
```

```
private void Button1_Click()
```

```
{
    uname = TB1.Text;
    password = TB2.Text;
```

// call the event

```
Login_Click();
```

```
}
```

```
}
```

Build → Build soln

go to form1.cs [Design] → drag and drop Login control from toolbox.

drag and drop progress bar and timer control from the toolbox.

Timer Control: This control is used to perform some operations repeatedly again and again after certain time interval.

Properties:-

1. Enabled: True/False: If true, timer control will work.
2. Interval: This property is used to set the time interval in milliseconds.

Events:-

Tick: This event will fire when timerenabled = true after certain time interval.

Progress Bar: This control is used to show the progress of o/p.

Properties:-

Value: This property is used to set the value of the progress bar control.

Minimum value is 0 and maximum value is 100.

goto form1.cs [Design] → select login control → properties → events → d.c on login_click & write the code:

```
private void login_Click()
{
    if(login.uname == "anil" && login.password == "anil")
    {
        timer1.Enabled = true;
    }
    else
    {
        MB.show("Invalid user");
    }
}
```

Select timer control → properties → events → d.c on tick event & write the code:

```
private void timer1_Tick()
{
    progressBar1.Value = progressBar1.Value + 5;
    if (progressBar1.Value >= 100)
    {
        timer1.Enabled = false;
        Success s = new Success(login.uname);
        s.show();
    }
}
```

Display the data in Data Gridview control:-

goto → form1.cs [Design] → Drag & drop DataGridView control from toolbox.

goto → form1.cs & write the code.

```
using System.Data.SqlClient;
```

```
private void form1_Load()
```

```
{
```

```
    SqlConnection con = new SqlConnection();
    SqlDataAdapter da = new SqlDataAdapter("select * from Employee", con);
    DataSet ds = new DataSet();
    da.Fill(ds, "Employee");
    dataGridView1.DataSource = ds.Tables[0];
}
```

↓
(or) tablename

performing the editing operations ^{with} in datagridview control.

goto → form1.cs [Design] → select datagridview control → select ellipse button → edit columns → Add → Name = edit
type = DataGridViewbutton column, header text = edit → add.

Select edit → properties → text = edit use column text for button = true →
goto → form1_Load and write the previous code.
press f5 and check the o/p.

Edit button will be displayed in each and every row.

whenever user clicks on edit button

→ Count the no. of rows that are available within the ^h
datagridview control.

- Rows property will count the no. of rows within the datagridctrl.
- Catch the index value of the row where user clicks on edit button.

e.RowIndex will catch the index value of the row.

→ Catch the cell based on which we want to perform editing operation.

for ex,

eno	ename	designation	sal
edit	103	Waseem	developer
30000			

Here, each and every cell can be identified by using index number.

Always the index number starts with 0.

cells[0] = edit, cells[1] = 103, cells[2] = Waseem, cells[3] = developer, cells[4] = 30000

So, here we are updating the employee information based on cells[1].

→ Catch the value i.e., that available within the cell.
→ update the record.

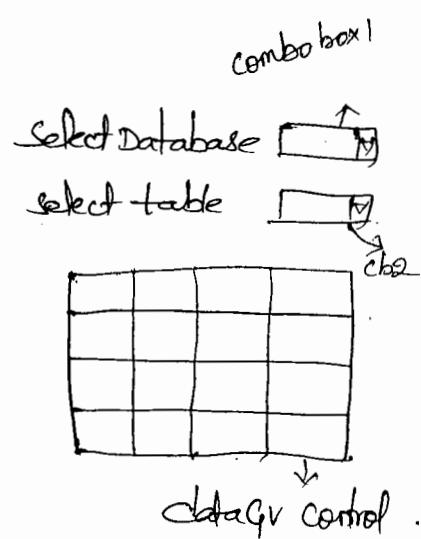
whenever user clicks on edit button, cell content click event will fire.
select DataGridView control → properties → events → d.c on cell
content click event and write the code:

```
using System.Data.SqlClient;  
private void DataGridView1_CellContentClick()  
{  
    int eno = Convert.ToInt32(DataGridView1.Rows(e.RowIndex).Cells[1].  
        Value);  
    string ename = DataGridView1.Rows(e.RowIndex).Cells[2].Value.  
        ToString();  
    string desig = DataGridView1.Rows(e.RowIndex).Cells[3].Value.ToString();  
    double sal = (DataGridView1.Rows(e.RowIndex).Cells[4].Value);  
    con.open();  
    string query = "update employee set eno=" + eno + "ename=" +  
        ename + "  
        desig=" + desig + ", sal=" + sal + ";";  
    SqlCommand cmd = new SqlCommand(query, con);  
    cmd.ExecuteNonQuery();  
    con.Close();  
}
```

Requirement:- Display the list of database
names in combobox1.

Display the list of table names that are
available within the selected database.

Display the data from selected table in
DataGridView control.



goto → form1.cs[Design] → design.

click on combobox write the code in -form1_Load.

private void form1_Load()

{

sql connection con = new sql connection();

con.open();

sql command cmd = new sql command ("select * from sys.databases", con);

sql datareader dr = cmd.ExecuteReader();

if (dr.HasRows)

{

while (dr.Read())

{

combobox1.Items.Add(dr[0]);

}

}

con.Close();

}

private void combobox1_SelectedIndexChanged()

{

combobox2.Items.Clear();

sql conn con = new sql conn();

con.open();

sql command cmd = new sql command ("select * from sys.Tables", con);

sql datareader dr = cmd.ExecuteReader();

if (dr.HasRows)

{

while (dr.Read())

{

combobox2.Items.Add(dr[0]);

```

}
}

con.Close();
}

private void comboBox2_SelectedIndexChanged()
{
    sqlConnection con = new SqlConnection("id=sa;pwd=abc;db=" + comboBox1.
    con = "selectedItem.ToString()"; dataSource = servername),
    SqlDataAdapter da = new SqlDataAdapter("Select * from comboBox2.
    selecteditem.ToString()", con);
    dataset ds = new dataset();
    da.Fill(ds);
    data GridView1.DataSource = ds.Tables[0];
}

```

performing deleting operation within the GridView control:-

goto → form1.cs [Design] → drag and drop Grid control from toolbox.
 goto → form1.cs & write the code:

```

private void form1_Load()
{
    sqlconnection con = new sqlconnection();
    sqlDataAdapter da = new sqlDataAdapter ("Select * from Employee", con);
    dataset ds = new dataset();
    da.Fill(ds, "Employee");
    dataGridView1.DataSource = ds.Tables[0];
}

```

Select ^{data} gridview control → select ellipse button → Edit columns
 → Add → name = delete type = Data gridView button column,
 or text = button → Add

Select → delete → properties → text = delete use column text for
button = true → OK.

go to → form_Load & write the previous code:

delete button will be displayed in each and every row.

whenever user clicks on delete button, cell content click event will fire.

Select DataGridView control → properties → events → d.c on cell
content click event and write the code:

using system.Data.SqlClient;

private void datagridview1_CellContentClick()

{

int eno = Convert.ToInt32(datagridview1.Rows(e.RowIndex).Cells[1].
value);

string ename = datagridview1.Rows(e.RowIndex).Cells[2].value.ToString();

string desig = datagridview1.Rows(e.RowIndex).Cells[3].value.ToString();

double sal = Convert.ToDouble(datagridview1.Rows(e.RowIndex).Cells[4].
value);

Con.open();

string query = "delete from employee where eno = " + eno + ",

sql Command cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

con.Close();

}

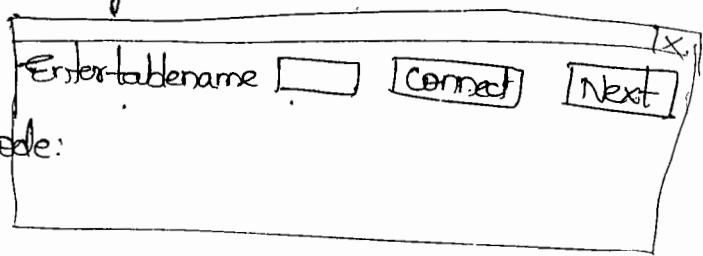
Generating the ~~user~~ controls dynamically:-

168
177

go to → form1.cs [Design]

go to → form1.cs and write the code:

using system.Data.SqlClient;
public partial class form1 : form



```
sql Connection con = new sqlconnection();
sql command cmd;
sql datareader dr;
Label[] L;
TextBox[] T;
int y = 50;
private void Button1_Click()
{
    con.open();
    string query = "select * from " + TB.Text + ";
    cmd = new sql command(query, con);
    dr = cmd.ExecuteReader();
    MessageBox.show(dr.fieldCount.ToString());
    L = new Label[dr.fieldCount];
    T = new TextBox[dr.fieldCount];
    for (int i = 0; i < dr.fieldCount; i++)
    {
        L[i] = new Label();
        L[i].Text = dr.GetName(i);
        L[i].Location = new point(100, y);
        this.controls.Add(L[i]);
    }
}
```

```

T[i] = new TextBox();
T[i].Location = new Point(200, y);
this.Controls.Add(T[i]);
y = y + 30;
}
}

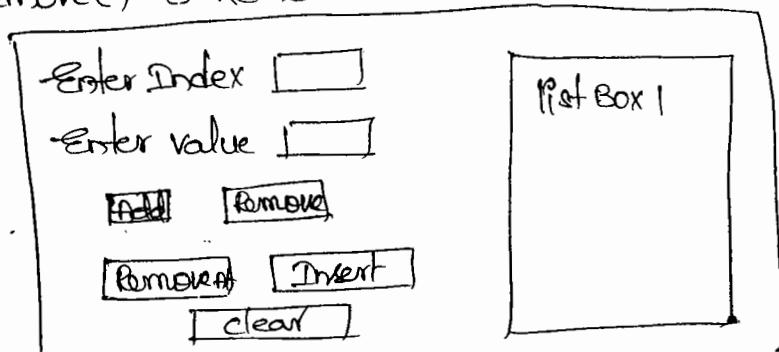
private void button1_Click()
{
    if (dr.Read())
    {
        for (int i = 0; i < dr.FieldCount; i++)
        {
            T[i].Text = dr[i].ToString();
        }
    }
    else
    {
        MessageBox.Show("no more records");
    }
}

```

ListBox control:- This control is used to add multiple items.

Methods:-

1. Add()
2. Remove()
3. RemoveAt()
4. Insert()
5. Clear()



D.C on Add button and write the code:

```
private void Button1_Click()
{
    listBox1.Items.Add(TB2.Text);
}

private void Button2_Click()
{
    listBox1.Items.Add(TB2.Text);
}

private void Button3_Click()
{
    listBox1.Items.RemoveAt(int.Parse(TB1.Text));
}

private void Button4_Click()
{
    listBox1.Items.Insert(int.Parse(TB1.Text), TB2.Text);
}
```

```
private void Button5_Click()
```

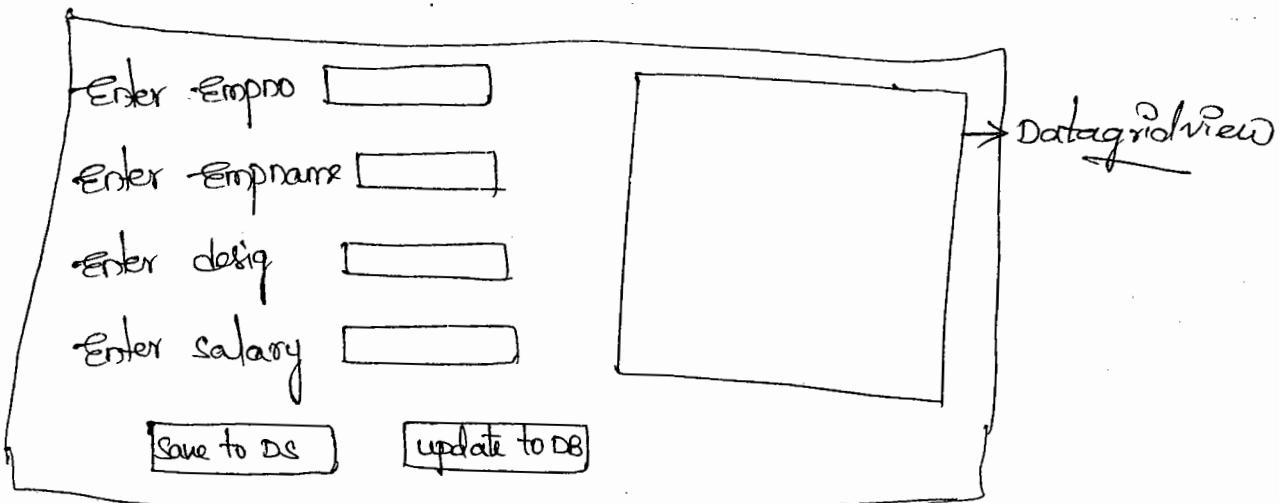
```
{
    listBox1.Items.Clear();
}
```

Working with sql command builder class:-

goto → sql ms studio

go to form1.cs [design]

empno	ename	deptno	sal



go to → project → add windows form → select app'config' file name =
 app.config → add

<configuration>

<appSettings>

<add key="constr" value="user id=sa; password=abc; d.b=employe;
 data source = "server name">

</appSettings>

</configuration>

go to → form1.cs and write the code

using system.data;

using system.data.sqlclient;

using system.configuration;

public partial class form1 : form

{

dataset ds;

datatable dt;

data row dr;

private void form1_Load()

{

// Create a dataset

ds = new dataset();

```
// create a Datatable  
dt = new DataTable();  
// assign the name for datatable  
dt.TableName = "emp";  
// create the datacolumns  
Datacolumn dc = new Datacolumn("empro");  
" dc1 = " ("ename");  
" dc2 = " ("desig");  
" dc3 = " ("Sal");  
// add the data columns to datatable  
dt.Columns.Add(dc);  
" " (dc1);  
" " (dc2);  
" " (dc3);  
}
```

```
private void Button1_Click()  
{  
// create a new row  
dr = dt.NewRow();  
// insert the records in datarow  
dr[0] = int.parse(TB1.Text);  
dr[1] = TB2.Text;  
dr[2] = TB3.Text;  
dr[3] = Double.parse(TB4.Text);  
// add the datarow to the datatable  
dt.Rows.Add(dr);  
// merge the datatable to dataset  
ds.Merge(dt);
```

```
dr.Delete();
dataGridView1.DataSource = ds.Tables[0];
MessageBox.Show("Record is updated to dataset");
}

private void button1_Click()
{
    SqlConnection con = new SqlConnection(ConfigurationSettings.AppSettings["constr"].ToString());
    SqlDataAdapter da = new SqlDataAdapter("Select * from emp", con);
    SqlCommandBuilder cb = new SqlCommandBuilder(da);
    da.Update(ds, "emp");
    MessageBox.Show("Record is updated to database");
}
```

Ex. program :-

goto → form1.cs [Design]

write the code in Button1_Click
protected void Button1_Click(\rightarrow int i)

```
    Listbox.Items.Add(Listbox.SelectedItem.Text);
```

Listbox1.Items.Remove(Listbox1.SelectedItem);

۳

Buttons.click()

۷

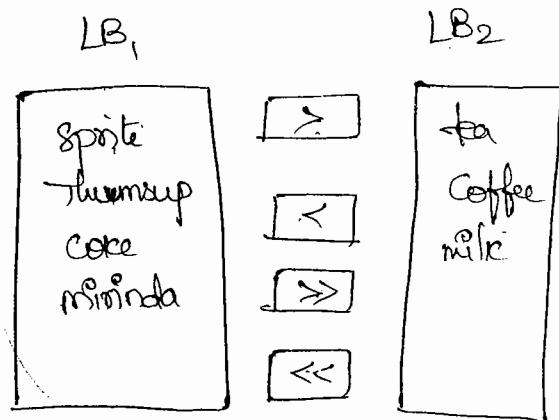
```
{  
    listBox1.Items.Add(Listbox2.SelectedItem.ToString());  
}
```

Listbox1.Items.Add(List1

Listbox1.Items.Remove(Listbox1.SelectedItem);

Listbox Items . Remove (listbox item , 2 , 3)

3



Button3_Click()

{

for (int i=0; i < PictureBox1.Items.Count; i++)

{

PictureBox2.Items.Add (PictureBox1.Items[i]);

}

PictureBox1.Items.Clear();

}

Button4_Click()

{

for (int i=0; i < PictureBox2.Items.Count; i++)

{

PictureBox1.Items.Add (PictureBox2.Items[i]);

}

PictureBox2.Items.Clear();

}

Any real time appⁿ was developed based on some architecture.
Every appⁿ is divided into two parts.

1. front end appⁿ
2. Back end appⁿ.

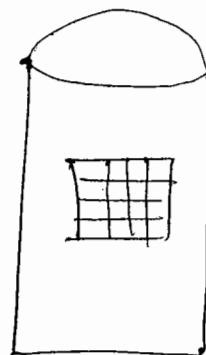
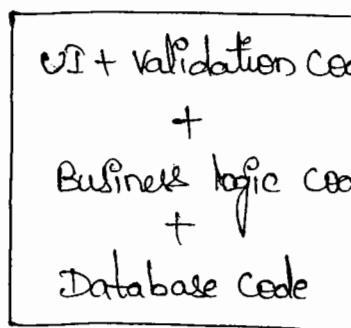
→ front end appⁿ can be developed by using c#.net (or) Asp.net.

Ex: windows forms appⁿ, console appⁿ, mobile appⁿ, web appⁿ.

→ Back end appⁿ can be developed by using MS SQL Server, Oracle etc.

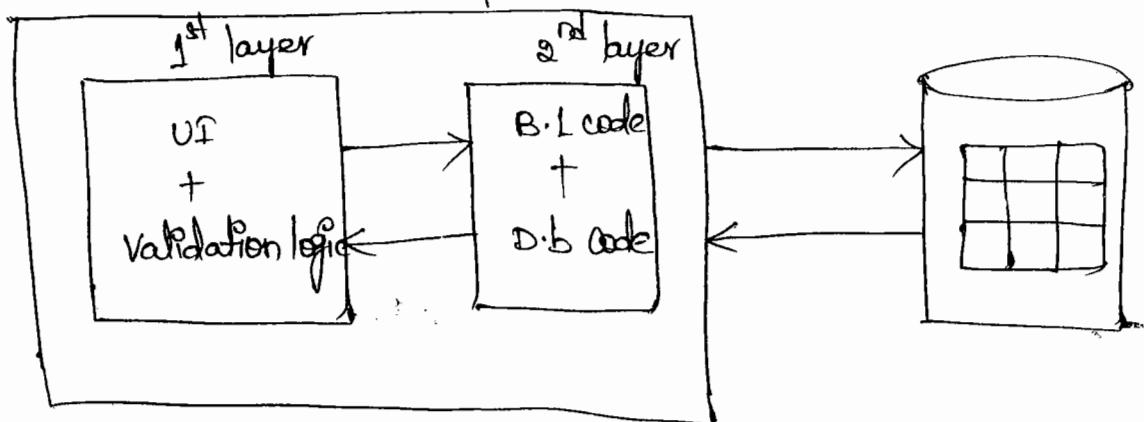
1-Tier Architecture (or) 1 Layered Architecture:-

In 1-Tier Architecture we will define the user interface code and validation logic, Business logic^{code} and data base code in a single file.



2-Tier Architecture:-

In 2-Tier Architecture we will define the user interface and validation logic in a separate file, Business logic code and database code in a separate file.



- In two tier architecture, the f.e appⁿ is divided into two layers.
- first layer consists of user interface + validation code.
- This code is available in Default.aspx and Default.aspx.cs
- second layer consists of Business logic code + Database code.
- The second layer can be developed by using class library.
- In 2-Tier architecture the first layer will communicate with second layer and second layer will communicate with database.

3-Tier Architecture:- In 3-Tier Architecture the f.e appⁿ is divided into 3 layers.

1. presentation layer
2. Business logic layer
3. Data access layer.

Presentation layer:- This layer consists of the user interface code and validation logic.

→ The U.I code can be written in form1.cs [source]

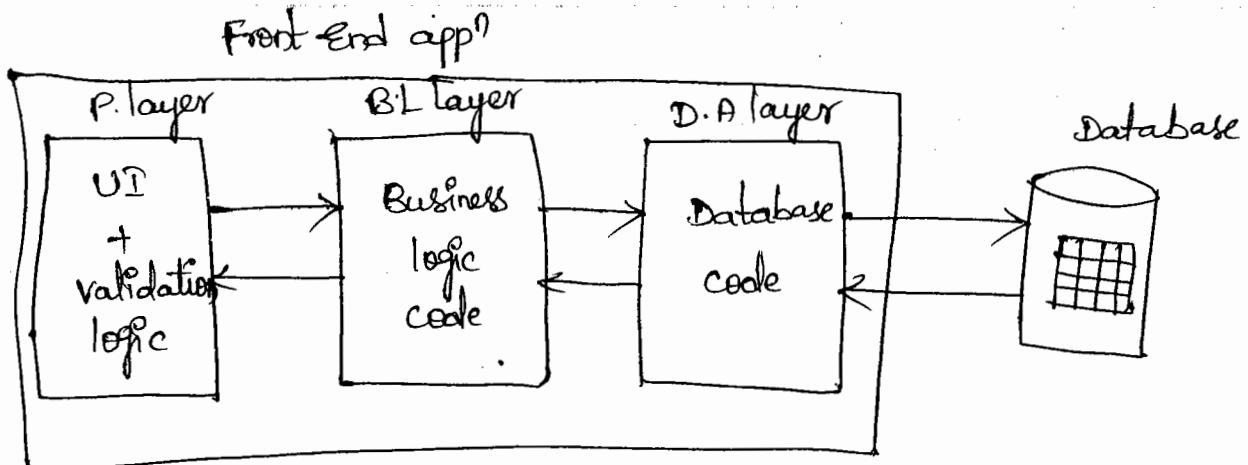
→ The validation logic can be written in form1.cs (or) Default.aspx.cs

Business logic layer:- This layer consists of the business logic code.

→ The B.L layer can be written in class library template.

Data access layer:- This layer consists of database related code.

→ The Data access layer can be written in class library template.



In 3-Tier Architecture the presentation layer will communicate with B.L layer and B.L layer will communicate with D.A layer and D.A layer will communicate with database.

Imp. Questions

1. what is CLR?

⇒ CLR is Common Language Runtime which is used for memory management and debugging.

⇒ CLR is the virtual machine component of Microsoft's .Net f/w.

2. what is CTS?

⇒ CTS is Common Type System which is used to provide common datatypes for all the .Net supportable languages.

3. what are nullable datatypes?

⇒ Nullable datatypes are used to assign a null value for integer datatype. It was introduced in .Net 2.0 & above versions.

Syntax: `int? a = null;`

4. what is the default datatype to store integer value?

⇒ `int`.

5. what is the default datatype to store decimal value?

⇒ `double`.

6. what is the diff b/w `Tostring()` & `Convert.ToString()` method?

⇒ `Tostring()` method doesn't handle null values, `Convert.ToString()` method will handle null values.

7. what is the diff b/w `Int.Parse()` and `Convert.ToInt32()` method?

⇒ `Int.Parse()` method is used to convert string to `int`, `Convert.ToInt32()` is used to convert any datatype to `int`.

8. what is the diff b/w value type D.T and reference type D.T?

⇒

Value type

1. Value type ref datatype are stored on stack.
2. Value type store real data
3. when passed as value type new copy is created and passed so changes to variable does not get reflected back.
4. Value types are faster in access.
5. Value types derive from system. value type.
6. V.T consists of primitive d.t.s, enumerators, structures.
7. V.T cannot contains the value null.

Reference type

1. Reference type d.t are stored on heap.
2. Reference type store reference of data.
3. when passed as reference type then reference of that variable is passed so changes to variable does reflect back.
4. Reference types are slower in access.
5. Reference types derive from system. object.
6. R.T consists of class, array, interface, delegates.
7. can contains the value null.

q. what is checked block and unchecked block?

⇒ checked block is used to enable overflow checking for arithmetic and conversion functions.

Ex: byte b = 255;

b++; o/p: 0 because the range of byte is 0-255
c.WL(b); so if the range is exceeded then the value of byte will round to zero.

But this type of programming is not safe. so, if the range of byte is exceeded then runtime error must occur.

Hence, we have to go for checked block.

```
checked  
{  
    byte b=255;  
    b++;  
    c.WL(b);  
}
```

O/P : A runtime error will occur saying that arithmetic operation resulted in an overflow.

10. write a program to get the range of byte.

⇒ c.WL(byte.MinValue);
c.WL(byte.MaxValue);

11. what is the diff b/w type.of() and sizeof()?

⇒ typeof() will get the base datatype name whereas sizeof() will get the size of the datatype.

12. what is the diff b/w Boxing and unboxing?

⇒ Boxing is a process of converting value type d.t to reference type d.t.

⇒ Unboxing is a process of converting reference type to v.t d.t.

13. what is widening and narrowing?

⇒ Widening:- It is a process of converting smaller datatype to longer datatype.

Ex: byte to int, int to long etc.

```
int i=10;      widening doesn't require explicit type casting.  
long l;  
l=i;  
c.WL(l);
```

Narrowing:- It is a process of converting longer d.t to smaller d.t.

Narrowing is unsafe type of programming because the data accuracy is not maintained.

while working with Narrowing, we have to do explicit typecasting.

Ex: long l = 10;
int i;
i = (int)l;
C.WL(i);

* 14. what is the diff b/w string and stringbuilder?

→ Both string and stringbuilder are classes. string is immutable and stringbuilder is mutable.

immutable means the value will not change.

mutable means the value will change.

string will allocate a new memory whenever we concatenate the string value but stringbuilder class will have a method 'append()'. This method is used to insert the new value on the existing value.

so, the usage of stringbuilder is more efficient in case of large amount of string manipulations.

If 100% of memory was allocated for a project then 98% of memory will be occupied by string but stringbuilder will occupy nearly 2% of memory.

so, performance wise stringbuilder is more efficient than string.

Ex. for string:

class program

{

static void Main()

{

String x = "";

for (int i=0; i<2; i++)

{

x = "sathya" + s;

c.WL(x);

c.WL(x.GetHashCode());

}

}

}

Ex. for stringbuilder:

stringbuilder x = new stringbuilder();

for (int i=0; i<2; i++)

{

x.append ("sathya" + x);

c.WL(x);

c.WL(x.GetHashCode());

}

15. what is an Assembly?

Assembly is the compiled format of any .net program which may be .dll (or) .exe.

16. How to View an Assembly?

goto → start → msvs2010 → msvs2010 → visual studio tools → visual studio command prompt → ILDASM.exe (Intermediate Language Disassembler).

17. what is the diff b/w private assembly and public assembly?

private Assembly:- The assembly that was specific for a single application is called as private assembly (or) folder specific assembly. A separate copy of dll will be copied in each and every location where we consumed the dll.

public Assembly:- The assembly that was registered under GAC is called as public assembly.

while working with public assembly a single copy of dll is maintained under GAC location.

18. what is GAC?

Global Assembly Cache

19. what is Reflection?

20. How to implement reflection in .net?

By using system.Reflection

system.Type

21. what is satellite assembly?

A satellite assembly is a .NET FW assembly containing sources specific to a given language. Using satellite assemblies, you can replace resources for different languages in different assemblies and the correct assembly is loaded into the memory only if the user selects to view the application in that language.

Satellite assembly is used to develop multilingual applications in .NET.

22. what are multilingual applications?

The applications that are built in supportive of more than one human readable language are known as multilingual applications.

23. what is the diff b/w $a=b$, $a==b$, $a.Equals(b)$

$a=b$ assigning b to a.

$a==b$ comparing a with b.

$a.Equals(b)$ Comparing objects.

24. what is the use of code snippets?

25. Diff b/w Array and Collection.

Array

Collection

1. Array is of fixed size
2. Array is the collection of values of the same datatype.
3. The variables in an array is called array elements.

1. ArrayList is allowed to grow as needed.
2. ArrayList is the collection of objects of different datatypes.

26. what is a class?

class is a keyword which is used to achieve encapsulation

class is a user defined reference type data type which consists of variables and methods.

27. what is the diff b/w class and structure?

class: class is reference type where structure is value type.

The memory for the members of class will be allocated on heap memory whereas the memory for the members of structure will be allocated on stack memory.

class supports inheritance but structure won't support inheritance.

28. what is encapsulation?

29. what is data abstraction?

30. what is constructor?

31. Diff. b/w constructor and method.

32. what is the diff b/w private constructor and static con?

private con is used to initialize the values for instance variables
static " " " " " static variables.

we cannot create an object for the constructor having private constructor outside the class.

we can create an object for the constructor having static constructor outside the class.

33. When the static constructor will be called?

34. Can we declare static access specifiers for static constructor?
no, because the call to the static constructor is done by CLR.

35. Can we declare parameterized static constructor?

no.

36. Can we override properties?

yes, using get accessor.

37. What is

38.

39. How to call default constructor

40. How to call default constructor of base class with parameterized constructor of base class?

base()

41. Does C# support multiple inheritance?

no, through interfaces we can achieve

42. What is overloading?

43. What is overriding?

44. How to restrict a class from inheritance?

Sealed.

45.

46. what is a delegate?

47. what is the diff b/w event and method?

method will have return type and event will not have return type.

48. what are Generics?

Generics are general datatypes which are used to avoid unnecessary boxing and unboxing & to avoid overloading.

49. Does generics support arithmetic operators?

No.

50. Can we use properties in interface?

passing parameters mechanism:-

1. pass by value
2. pass by reference
3. out mode
4. params mode.

pass by value:- In this mode we will pass parameter as values
the changes on one variable will not affect the value of
another variable.

class program

```
{  
public static void show(int j)  
{ j=100; } }
```

O/P : 10

static void Main()

```
{  
int x=10;  
show(x);  
Console.WriteLine(x);  
}
```

x → 10

j → 100

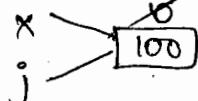
Observation:- In the above program, i and j are pointing to different memory locations so the operations on one variable will not change the another variable.

In pass by value mechanism the actual parameters will not affect formal parameters.

Actual parameters: The values that we pass show(x).

formal parameters: method parameters show(int j)

pass by reference:- In this mode both the variables are pointing to same memory location so, the operations on one variable will affect the value of another variable.



so, here the actual parameters will vary with formal parameters.

Ex: class program

```
{  
public static void show(ref int j)  
{  
    j = 100;  
}  
static void Main()  
{  
    int x=10;  
    show( ref x);  
    c.WL(x);  
}
```

O/P : 100

out mode:- It is used when a method wants to return more than one value i.e., when we want to return more than one class program value from a function then we have to use out mode.

```
{  
public static void show(int x, int y, out int sum, out int product)  
{
```

```
sum = x+y;
```

```
product = x*y;
```

```
}
```

```
static void Main()
```

```
{
```

```
int total;
```

```
int p;
```

```
show(10, 20, out total, out p);
```

```
c.WL ("Sum is "+total);
```

```
c.WL ("product is "+p);
```

```
} }
```

params mode:- If we want to make parameters optional then we have to go for params mode.

Note:- A parameter array must be the last parameter in formal parameter list.

Ex: class program

```
{
```

```
public void show(params int[] numbers)
```

```
{
```

```
c.WL (numbers.Length);
```

```
foreach (int i in numbers)
```

```
{
```

```
c.WL (i);
```

```
} }
```

```
static void Main()
```

```
{ int[] Ar = new int[5];
```

```
Ar[0] = 10;
```

```
Ar[1] = 20;
```

```
Ar[2] = 30;
```

```
Ar[3] = 40;
```

```
Ar[4] = 50; } }
```

Access specifiers

It will define the scope of the type as well as their members i.e., who can access and who cannot access.

→ Different Access specifiers are:

1. private
2. public
3. protected
4. internal
5. protected internal

private:- The scope of private is within the class. By default the members of the class are private i.e., we cannot access the private members of one class to another class.

Q) what will happen if we declare a constructor as private?
we cannot create an object outside the class.

public:- The scope of public is within the assembly (or) outside the assembly i.e., there is no res^d for public types & members.

protected:- The members that are declared as protected under a class can be accessed only within the class (or) child class.

→ Non child class cannot access them.

→ we cannot declare protected access specifiers for types, delegates, classes etc.

→ The scope of protected is within the class.

class program

{

protected int x;

static void Main()

✓

{

program p = new program();

p.x = 10;

c.wl(p.x);

} }

```
class A
{
protected int x;
}

class program
{
    static void Main()
{
```

```
    A a; = new A();
    a.x = 10;
    c.WL(a,x);
}
```

X [we cannot access protected member in another class]

→ the scope of protected is within the class (or) in the immediate derived class.

```
class A
{
protected int x;
}

class B : A
{
    static void Main()
{
    B b; = new B();
    b.x = 10;
    c.WL(b,x);
}
```

- Internal: - The scope of assembly is within the assembly.
 → The default access specifiers for type is internal like class, delegate, interface.
 → Internal is working like public within the assembly.
 → The scope of Internal is within the namespace.

using system;

namespace consoleapplication1

{
class A

{
internal void show()

{
c.wl("I am show()"); } }

class program

{
static void Main()
{
A a = new A();
a.show();
}

namespace n1

{
class A
{
internal int x;
}

using n1;

namespace n2

{
class program
{
static void Main()
{
A a = new A(); X
a.x=10;
}

protected internal: - The members declared as protected internal have dual scope i.e., within the project they behave like internal. outside the project they behave like protected.

```

namespace n1
{
public class A
{
protected internal int x;
}
}

```

```

namespace n2
{
class program1 A
{
static void Main()
{
A a;  
a.x=10;
}
}

```

1. within the class
2. within child class
of same project
3. within nonchild class
of same project
4. within child class of
another project
5. nonchild class of
another project

	<u>private</u>	<u>internal</u>	<u>protected</u>	<u>Friend</u>	<u>Pub</u>
1.	✓	✓	✓	✓	✓
2.	x	✓	✓	✓	✓
3.	x	✓	x	✓	✓
4.	x	x	x	✓	✓
5.	x	x	x	x	✓

Serialization: whenever an object is created, the data that is stored in the object will be maintained temporarily until the program is running (or) the data will be lost when the object is unreferenced.

→ once the object is destroyed, the data will be lost but if we want to use the same data for next time it is not possible. making object data permanent by storing into a file (or) database (or) to any other memory device is called as object persistency

→ once the object is persisted, that object data can be reused whenever required. for this to reuse the object data we're to convert the object into other streams. This process is serialization.

serⁿ: It is a process of converting an object into stream of bytes in order to persist it to memory (or) db (or) file.

Different types of serⁿ:

Binary serⁿ:

