

22/4/20

C# Methods

A method is a block of code which runs when it is called.

Methods are used to perform certain actions and they are also known as functions.

Methods similarly working as a function.

Syntax - like

```

static void xyz()
{
    // 
}

```

Method

Ex namespace

```

{
    class func
    {
        static void mymethod()
        {
            Console.WriteLine("Welcome dear");
        }

        static void main (string[] args)
        {
            mymethod(); // it's calling of method
        }
    }
}

```

Method Parameters

↳ Passing value

(2)
passing value.

Ex class xyz

```
{ static void mymethod (String frame)
```

```
{
```

```
    consider writing (frame);
```

```
}
```

```
static void main (String [] arg).
```

```
{
```

```
    mymethod ("keena");
```

```
    mymethod ("kicha");
```

```
}
```

```
} }
```

OP

— keena.
kicha.

default Parameter

(3)

In default Parameter, if we are not assign the value it auto fill itself full. fill the value in execution time.

Ex class Q42

```
{  
    static void mymethod (String value = 30)  
    {  
        console.WriteLine (value);  
    }  
}
```

```
static void main (String [] args)
```

```
{  
    mymethod (100);
```

```
    mymethod (40);
```

```
    mymethod (); ← function call without  
                    any value passer.
```

```
    mymethod (63);
```

```
}
```

```
}
```

Q/P

100
40
30 ← default
63

Multiple Parameters

(4)

```
class xyz
```

```
{  
    static void mymethod(String name, int age)  
    {  
        Console.WriteLine(name + " is " + age);  
    }  
}
```

```
static void main (String [] args)
```

```
{  
    mymethod("Gaurav", 20);  
    mymethod("Sandeep", 18);  
    mymethod("Rohan", 20);  
}
```

O/P Gaurav is 20
Sandeep is 18
Rohan is 20

return values

(5)

Ex

class XYZ

```
{  
    static void int MyMethod(int n){  
        {  
            return 5 * n;  
        }  
    }
```

```
static void main(String[] args){  
    {
```

MyMethod

```
        Console.WriteLine(MyMethod(8));  
    }
```

```
}
```

OP — 40 (5*8)

```
static int MyMethod(int x, int y)
```

```
{ return x * y;  
}
```

```
}
```

```
static void main(String[] args)
```

```
{ int z = MyMethod(10, 20);  
    Console.WriteLine(z); }
```

```
}
```

Named Arguments

(6)

class K12

```
static void myMethod (String c1, String c2)  
{  
    Console.WriteLine ("The youngest child is", c2);  
}
```

```
static void mymain (String [] args)  
{  
    myMethod (c1: "Anil", c2: "Rahul");  
}  
}
```

Q The youngest child is Rahul.

Method overloading

(7)

it means multiple method can have the same name with different parameters.

class X42

```
{ static int PlusMethod (int n, int y) .  
  { return n+y ;  
  }
```

```
static double PlusMethod (double x,  
                           double y )  
{  
  return n+y ;  
}
```

```
{
```

```
static void main (String [] args)
```

```
{ int N1 = PlusMethod (10, 20) ;  
  double N2 = PlusMethod (5.2, 2.5) ;
```

```
  Console.WriteLine ("int sum = " + N1) ;
```

```
  Console.WriteLine ("double sum = " + N2) ;  
}
```

```
} output = 30
```

```
< 7.7 .
```

22/4/20

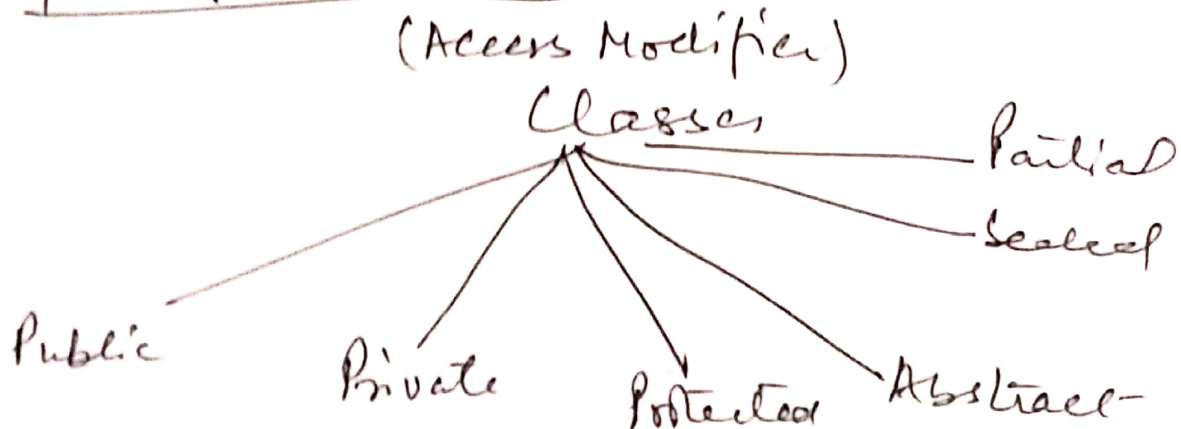
Class

Class is user defined data type with a template that serve to define the properties.

Classes is collection of member variable and member function on the basis of object.

```
class classname
{
    {variable declaration',
    {method declaration', }
}
```

Types of classes



Class and Object

(18)

Class and objects are the two main aspects of object oriented programming.

<u>Like</u>	Class	Object
	Fruit	Apple Mango Banana

or

Car	Hyundai Volvo Honda
-----	---------------------------

a class is a template for objects and
an object is an instance of a class.

Ex

```
class Car
{
    string color = "Blue";
    static void main (String[] args)
    {
        Car myobj = new Car();
        Console.WriteLine(myobj.color);
    }
}
```

class name →

object

it's same as

Car.cs

Car.cs

or Blue.

C# class Members

(3)

```
Ex class X42
{
    string color = "red";
    int maxspeed = 200;
}

static void main (String[] args)
{
    X42 K1 = new X42();
    Console.WriteLine(K1.color);
    Console.WriteLine(K1.maxspeed);
}

// Output:
// red
// 200
```

datatype
initialization

Multiple Objects -

class Car

```
{
    string model;
    string color;
    int year;
}
```

static void main (String[] args)

```
{
    Car Volvo = new Car();
    Volvo.model = "Bengaluru";
    Volvo.color = "Blue";
    Volvo.year = 1969;
}
```

```
Car Opel = new Car();
Opel.model = "Mustang";
Opel.color = "red";
Opel.year = 2005;
```

```
Console.WriteLine
(Volvo.model);
Console.WriteLine
(Opel.model);
}
```

Object Methods

(10)

Ex class stud

```
{ string branch; // field  
  int year; // field  
  public void present() // method  
  {  
    console.WriteLine("The student is not  
                        present today");  
  }
```

```
static void main(string[] args)
```

```
{  
  stud myObj = new stud();  
  myObj.present(); // calling of method.  
}  
}
```

Op - The student is not present today.