Object Oriented Strategy : Encapsulation

There are four Object Oriented Paradigms of C# as
1. Encapsulation
2. Abstraction
3. Polymorphism
4. Inheritance

Encapsulation means binding characteristics and behaviour together.

When we design any class the data members are called as characteristics and methods in that class are called as behaviours.

Application 1 :

- Application which demonstrates concept of class.
- In this application we design one class named as Employee which contains three characteristics and two behaviours.
- Inside main method we create two objects of that class.
- When we create object memory for its characteristics gets allocated separately.
- Then we can invoke the methods using object.

```
using System;

class Employee
{
        public int Eid;
        public String Ename;
        public int Esalary;

        public void Accept()
        {
                // Code
        }

        public void Display()
        {
                // Code
        }
}

class Program
{
        static void Main(string[] args)
        {
                Employee eobj1 = new Employee();
                Employee eobj2 = new Employee();

                eobj1.Accept();
                eobj1.Display();
        }
}
```

Application 2 :
Application which demonstrates concept of this keyword.

- In C# when we call any non static method the reference of caller object gets created implicitly by the compiler.
- As a programmer we can use this keyword to access every non static member of our caller object.
- We can use this keyword in constructor, destructor, non static method.

```
using System;

class Employee
{
	public int Eid;
	public String Ename;
	public int Esalary;

	public Employee(int value, String name, int sal)
	{
		this.Eid = value;
		this.Ename = name;
		this.Esalary = sal;
	}

	public void DisplaySalary()
	{
		Console.WriteLine("Salary is {0}",this.Esalary);
	}

	public void Display()
	{
		Console.WriteLine("Employee name is {0}",this.Ename);
		Console.WriteLine("Employee id is {0}",this.Eid);
		this.DisplaySalary();
	}
}

class Program
{
	static void Main(string[] args)
	{
		Employee eobj1 = new Employee(11,"Amit",25000);
		Employee eobj2 = new Employee(21,"Sumit",42000);

		eobj1.Display();
		eobj2.Display();
	}
}
```