**Virtual Class V/S Interface V/S Abstract Class**

**Virtual Class**

**When a Class have atleast one Virtual Method then class is called Virtual Class.**

**Example of Virtual Class:**

**class Vehical**

**{**

**public string Name;**

**public string VehicalName(string name)**

**{**

**Name = name;**

**return Name;**

**}**

**public virtual void VehicalTires() {**  //Virtual Method of Virtual Class “vehical”

**Console.WriteLine("Showcase of Vehical Tires");**

**}**

**}**

**class Truck : Vehical**

**{**

**public string VehicalName(string name)**

**{**

**return base.VehicalName(name);**

**}**

**public override void VehicalTires()** //Override Method of Virtual Method “VehicalTires”

**{**

**Console.WriteLine("Truck has 6 Tires");**

**}**

**}**

**class Program**

**{**

**private static void Main(string[] args)**

**{**

**Truck t = new Truck();**

**Vehical vt = new Truck();**

**Vehical v = new Vehical();**

**Console.WriteLine(vt.VehicalName("Mahindra"));**

**vt.VehicalTires();**

**Console.WriteLine(t.VehicalName("AirIndia"));**

**t.VehicalTires();**

**v.VehicalTires();**

**}**

**}**

**Interface**

**A interface mast have atleast one abstract method and the interface cannot have any data members.**

**Example of Interface :**

**interface Vehical**

**{**

**public abstract string VehicalName(string name); //abstract method called**

**public virtual void VehicalTires() //Virtual method**

**{**

**Console.WriteLine("Showcase of Vehical Tires");**

**}**

**}**

**}**

**class Truck : Vehical**

**{**

**public string Name;**

**public string VehicalName(string name) //must be call with its body**

**{**

**Name= name;**

**return Name;**

**}**

**public void VehicalTires() //overridden Method of VehicalTires**

**{**

**Console.WriteLine("Truck has 6 Tires");**

**}**

**}**

**class Program**

**{**

**private static void Main(string[] args)**

**{**

**Truck t = new Truck();**

**Vehical vt = new Truck();**

**Console.WriteLine(vt.VehicalName("Mahindra"));**

**vt.VehicalTires();**

**Console.WriteLine(t.VehicalName("AirIndia"));**

**t.VehicalTires();**

**}**

**}**

**Abstract Class**

**When a class has only abstract methods and you can create data members but abstract methods have no body thats why there is no mandatory to create data members, then the class is called Abstract Class.**

**Example of Abstract Class :**

**abstract class Vehicle**

**{**

**public abstract string VehicalName(string name);** //abstract method

**public abstract void VehicalTires();** //abstract method

**}**

**}**

**class Truck : Vehicle**

**{**

**string Name;**

**public override string VehicalName(string name)**  //overridden method of abstract method

**{**

**Name = name;**

**return Name;**

**}**

**public override void VehicalTires()** //overridden method of abstract method

**{**

**Console.WriteLine("Truck has 6 tires");**

**}**

**}**

**class Program**

**{**

**private static void Main(string[] args)**

**{**

**Truck t = new Truck();**

**Console.WriteLine(t.VehicalName("Mahindra"));**

**t.VehicalTires();**

**}**

**}**