

Interface

```
interface IUser
{
    void Getdetails(string n);
}
interface IUserofficial
{
    void officialdetails();
}
class user:IUser,IUserofficial
{
    public string name = "riya";
    public void Getdetails(string n)
    {
        Console.WriteLine("Name:{0}", n);
    }
    public void officialdetails()
    {
        Console.WriteLine();
    }
}
class user1:IUser
{
    public void Getdetails(string n)
    {
        Console.WriteLine("Name:{0}", n);
        Console.WriteLine("Location:{0}", l);
    }
}
class Program
{
    static void Main(string[] args)
    {
        IUser u = new user();
        u.Getdetails("Reena");
        IUser u1 = new user1();
        u1.Getdetails("Bangalore");
    }
}
```

Abstract Class

```
abstract class info
{
    abstract public void getdetails(string n, string l, int a);
}
class user : info
{
    public override void getdetails(string n, string l, int a)
    {
        Console.WriteLine("Name:{0}", n);
        Console.WriteLine("location:{0}", l);
        Console.WriteLine("Age:{0}", a);
    }
}
class Program
{
    static void Main(string[] args)
    {
        user us = new user();
        us.getdetails("Ananya", "Chennai", 24);
    }
}
```

Abstract Class

```
abstract class laptop
{
    public string brand { get; set; }
    public string model { get; set; }

    public void laptopdetails()
    {
        Console.WriteLine("Brand:{0} and Model:{1}", brand, model);
    }
    abstract public void motherboardinfo();
}

class user : laptop
{
    public override void motherboardinfo()
    {
        Console.WriteLine("Motheboard Information");
    }
}
class Program
{
    static void Main(string[] args)
    {
        /* laptop lp = new laptop();
        lp.brand = "Dell";
        lp.model = "Intel5";
        lp.laptopdetails();*/
    }
}
```

```

        user u1 = new user();
        u1.brand = "Dell";
        u1.model = "Intel5";
        u1.laptopdetails();
        u1.motherboardinfo();
    }
}

```

Method Overrriding

```

class user
{
    public string name = "Suresh";
    public string location = "Hyderabad";

    public virtual void getinfo()
    {
        Console.WriteLine("Name: {0}", name);
        Console.WriteLine("Location: {0}", location);
    }
}
sealed class Details:user
{
    public int age = 25;

    public override void getinfo()
    {
        Console.WriteLine("Name: {0}", name);
        Console.WriteLine("Location: {0}", location);
        Console.WriteLine("Age: {0}", age);
    }
}
class officialdetails : user
{
    public int sal = 40000;

    public override void getinfo()
    {
        Console.WriteLine("Name: {0}", name);
        Console.WriteLine("Location: {0}", location);
        // Console.WriteLine("Age: {0}", age);
        Console.WriteLine("Salary: {0}", sal);
    }
}

class Program
{
    static void Main(string[] args)
    {
        officialdetails od = new officialdetails();
        od.getinfo();
    }
}

```

```
    }  
}
```

Method Overriding & Sealed

```
public class employee  
{  
    public int salary = 30000;  
  
    public virtual void total()  
    {  
        Console.WriteLine("Employee\'s Total Salary"+salary);  
    }  
}  
  
public class programmer : employee  
{  
    public int bonus = 10000;  
  
    public override void total()  
    {  
        Console.WriteLine("Employee\'s Total Salary" + salary);  
        Console.WriteLine("Total Salary for programmer is:" + (bonus  
+salary));  
    }  
}  
  
public sealed class developer : programmer  
{  
    public int bonus1 = 20000;  
    public override void total()  
    {  
        Console.WriteLine("Employee\'s Total Salary" + salary);  
        Console.WriteLine("Total Salary for programmer is:" + (bonus +  
salary));  
        Console.WriteLine("Total Salary for developer is:" + (bonus1 +  
salary));  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        // programmer p1 = new programmer();  
        // p1.total();//base class method  
        // p1.totbonus();//derived class method
```

```
        //Console.WriteLine(p1.salary + p1.bonus);

        developer d1 = new developer();//for the last derived class
        d1.total();
    }
}
```