

## EXPT 4

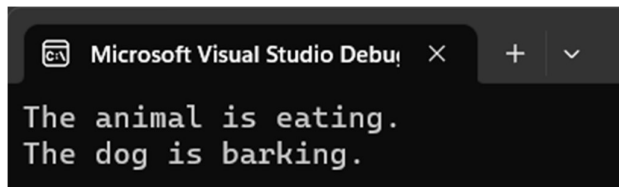
### 1) Single Inheritance

```
using System;
class Animal
{
    public void Eat()
    {
        Console.WriteLine("The animal is eating.");
    }
}
class Dog : Animal
{
    public void Bark()
    {
        Console.WriteLine("The dog is barking.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Dog myDog = new Dog();

        myDog.Eat();
        myDog.Bark();
    }
}
```

**Output: -**



## 2) Multiple Inheritance

```
using System;
interface IAnimal
{
    void Eat();
    void Sleep();
}
interface IRobot
{
    void TurnOn();
    void TurnOff();
}
class RobotDog : IAnimal, IRobot
{
    public void Eat()
    {
        Console.WriteLine("The robot dog is eating.");
    }

    public void Sleep()
    {
        Console.WriteLine("The robot dog is sleeping.");
    }

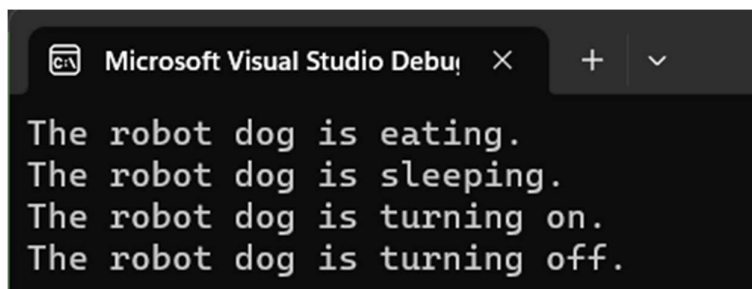
    public void TurnOn()
    {
        Console.WriteLine("The robot dog is turning on.");
    }

    public void TurnOff()
    {
        Console.WriteLine("The robot dog is turning off.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        RobotDog myRobotDog = new RobotDog();

        myRobotDog.Eat();
        myRobotDog.Sleep();
        myRobotDog.TurnOn();
        myRobotDog.TurnOff();
    }
}
```

**Output: -**

A screenshot of the Microsoft Visual Studio Debug Console. The window title is "Microsoft Visual Studio Debug Console" with a close button (X) and a dropdown arrow. The console output shows four lines of text: "The robot dog is eating.", "The robot dog is sleeping.", "The robot dog is turning on.", and "The robot dog is turning off.".

```
Microsoft Visual Studio Debug Console
The robot dog is eating.
The robot dog is sleeping.
The robot dog is turning on.
The robot dog is turning off.
```

### 3) Hybrid Inheritance

```
using System;

interface IAnimal
{
    void Eat();
    void Sleep();
}

interface IPlant
{
    void Photosynthesize();
}

interface IOmnivore : IAnimal, IPlant
{
    void Hunt();
}

class Omnivore : IOmnivore
{
    public void Eat()
    {
        Console.WriteLine("The omnivore is eating.");
    }

    public void Sleep()
    {
        Console.WriteLine("The omnivore is sleeping.");
    }

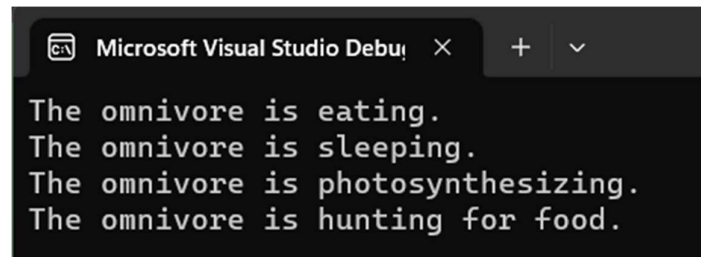
    public void Photosynthesize()
    {
        Console.WriteLine("The omnivore is photosynthesizing.");
    }

    public void Hunt()
    {
        Console.WriteLine("The omnivore is hunting for food.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Omnivore myOmnivore = new Omnivore();

        myOmnivore.Eat();
        myOmnivore.Sleep();
        myOmnivore.Photosynthesize();
        myOmnivore.Hunt();
    }
}
```

**Output: -**



```
Microsoft Visual Studio Debug Console
The omnivore is eating.
The omnivore is sleeping.
The omnivore is photosynthesizing.
The omnivore is hunting for food.
```