

EXPT 3

***Interface**

using System;

```
public class Animal // Base Class
{
    public void Eat()
    {
        Console.WriteLine("Animal is eating");
    }
}

public class Dog : Animal // Derived Class
{
    public void Bark()
    {
        Console.WriteLine("Dog is barking");
    }
}

public class Bulldog : Dog // Multilevel Inheritance
{
    public void Snore()
    {
        Console.WriteLine("Bulldog is snoring");
    }
}

public class Cat : Animal // Derived Class
{
    public void Meow()
    {
        Console.WriteLine("Cat is meowing");
    }
}

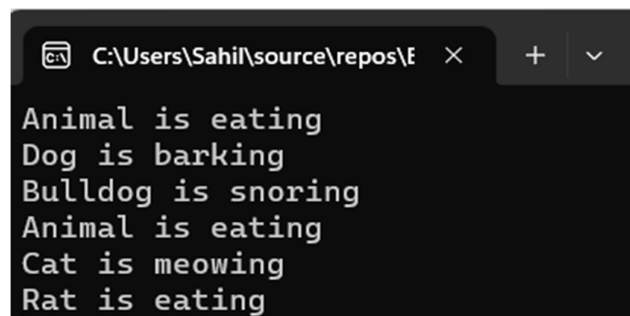
// Interface
public interface IAnimal
{
    void REat();
}

// Class implementing the interface
public class Rat : IAnimal
{
    public void REat()
    {
        Console.WriteLine("Rat is eating");
    }
}
```

```
}  
}
```

```
class Program  
{  
    static void Main(string[] args)  
    {  
        Bulldog b = new Bulldog();  
        b.Eat();  
        b.Bark();  
        b.Snore();  
  
        Cat c = new Cat();  
        c.Eat();  
        c.Meow();  
  
        Rat d = new Rat();  
        d.REat(); // Accessing interface method  
        Console.ReadLine();  
    }  
}
```

Output :-



```
C:\Users\Sahil\source\repos\F x + v  
Animal is eating  
Dog is barking  
Bulldog is snoring  
Animal is eating  
Cat is meowing  
Rat is eating
```

*Static, Non-Static Class

```
using System;  
  
public static class StaticClass  
{  
    public static void PrintMessage()  
    {  
        Console.WriteLine("This is a static class method.");  
    }  
}
```

```

    }
}

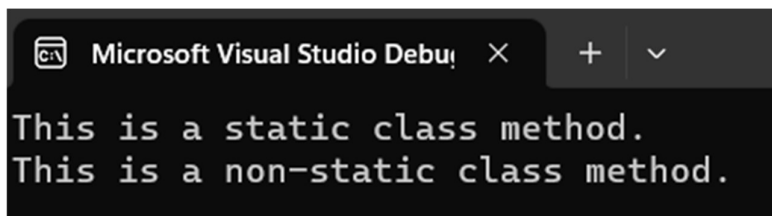
public class NonStaticClass
{
    public void PrintMessage()
    {
        Console.WriteLine("This is a non-static class method.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        StaticClass.PrintMessage();

        NonStaticClass nonStaticObj = new NonStaticClass();
        nonStaticObj.PrintMessage();
    }
}

```

Output: -



***Extension Method**

```

using System;

namespace ExtensionMethod
{
    class sahil
    {
        // Method 1
        public void M1()
        {
            Console.WriteLine("Method Name: M1");
        }

        // Method 2
        public void M2()

```

```

        {
            Console.WriteLine("Method Name: M2");
        }

        // Method 3
        public void M3()
        {
            Console.WriteLine("Method Name: M3");
        }
    }

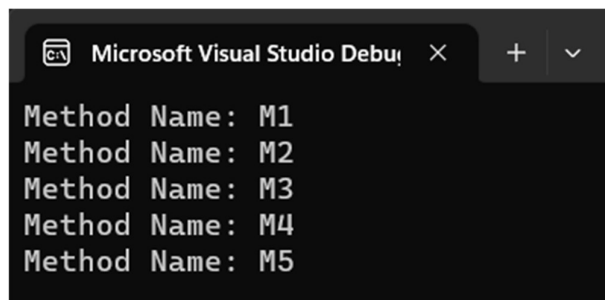
    static class NewMethodClass
    {
        // Method 4
        public static void M4(this sahil g)
        {
            Console.WriteLine("Method Name: M4");
        }

        // Method 5
        public static void M5(this sahil g, string str)
        {
            Console.WriteLine(str);
        }
    }

    public class sutar
    {
        public static void Main(string[] args)
        {
            sahil g = new sahil();
            g.M1();
            g.M2();
            g.M3();
            g.M4();
            g.M5("Method Name: M5");
        }
    }
}

```

Output: -



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output displays five lines of text, each on a new line: "Method Name: M1", "Method Name: M2", "Method Name: M3", "Method Name: M4", and "Method Name: M5".

```

Method Name: M1
Method Name: M2
Method Name: M3
Method Name: M4
Method Name: M5

```