**C# Code:**

#**Sample code to implement static method:**

using System;

class Program

{

static void staticmethod()

{

Console.WriteLine(“Static Method”);

}

static char staticmethod1()

{

Console.WriteLine(“Instance Method”);

return ‘C’;

}

static void Main()

{

Program.staticmethod();

Console.WriteLine(Program.staticmethod2());

programInstance.MethodB();

Console.Read();

}

}

OUTPUT:

Static Method

Static Method

C

Instance Method

**#Sample code to demonstrate pass by value parameter**

using System;

class program

{

static void Cube(int x)

{

x=x\*x\*x;

Console.WriteLine(“Value within the cube method : {0}”,x);

}

static void Main()

{

Int num=5;

Console.WriteLine(“Value before the method is called : {0}”,num);

Cube(num);

Console.WriteLine(“Value after the method is called : {0}”,num);

Console.ReadKey();

}

}

OUTPUT:

Value before the method s called : 5

Value within the cube method : 125

Value after the method is called : 5

#Sample code to illustrate method hiding

using System;

using System.Collection.Generic;

using System.Text;

namespace ConsoleApp

{

public class Demo

{

public virtual double Area(double r)

{

return r\*r;

}

public void func()

{

Console.WriteLine(“Base Class”);

}

}

public class A : Demo

{

public override double Area(double r)

{

return base.Area(r)\*r;

}

public new void func()

{

Console.WriteLine(“Derived Class”);

}

}

public class Test

{

public static void Main(string[] args)

{

A o1=new A();

Console.WriteLine(o1.Area(20));

O1.func();

Console.ReadLine();

}

}

}

OUTPUT:

8000

Derived Class