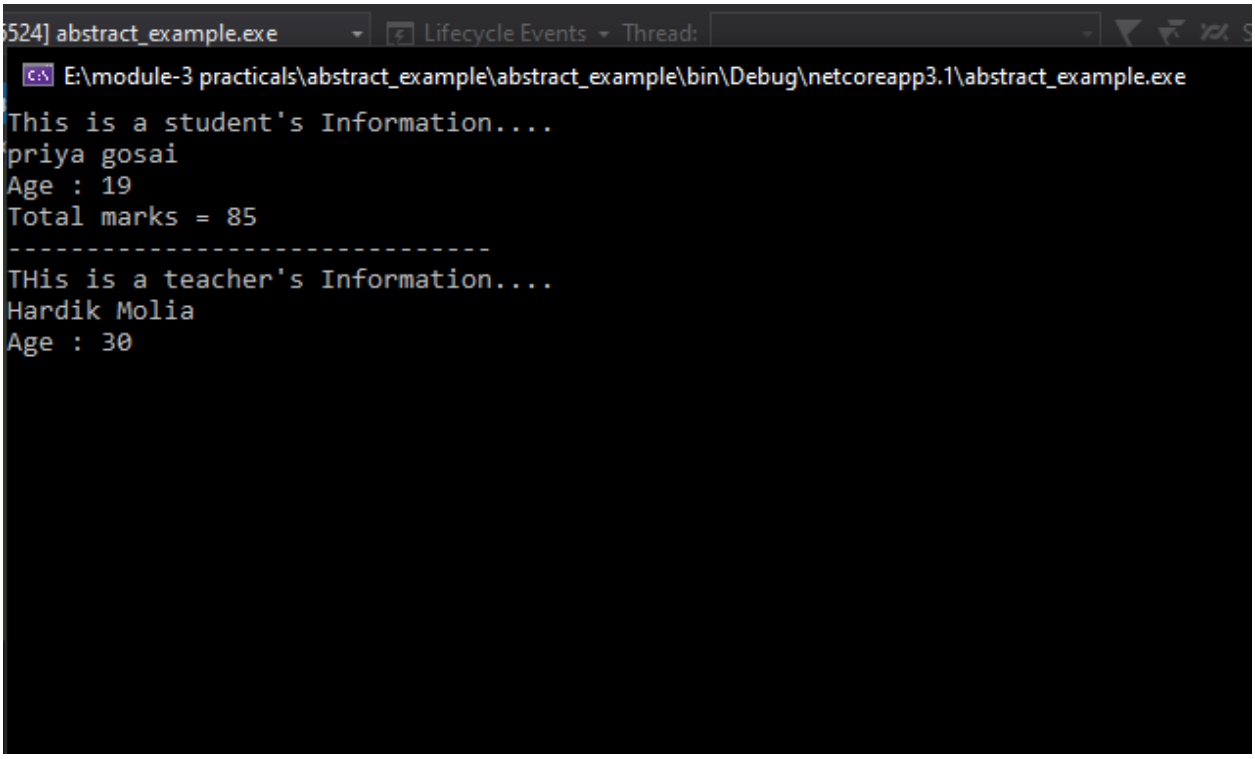
1. Abstract class example

```
using System;

    namespace abstract_example
    {
        abstract class School
        {
            public String fname;
            public String lname;

            public int age;

            public abstract void getfullname();
            public abstract void getage();
        }
        class Student : School
   {
            public int sub1_marks;
            public int sub2_marks;
            public int sub3_marks;
            public String subject;

            public override void getfullname()
            {
                Console.WriteLine("This is a student's Information....");
                Console.WriteLine(fname + " " + lname);
            }
            public override void getage()
            {
                Console.WriteLine("Age : {0}", age);
            }

            public void totalmarks()
            {
             int result = (sub1_marks + sub2_marks + sub3_marks) / 3;
                  Console.WriteLine("Total marks = {0}", result);
              }
          }
          class Teacher : School
          {
              public double salary;
              public String Subject;

        public override void getfullname()
       {
                  Console.WriteLine("This is a teacher's Information....");
```

```
            Console.WriteLine(fname + " " + lname);

        }
        public override void getage()
        {
            Console.WriteLine("Age : {0}", age);
        }



    }
    class Program
    {
        static void Main(string[] args)
{
      Student s1 = new Student();
            s1.fname = "priya";
            s1.lname = "gosai";
            s1.age = 19;
            s1.sub1_marks = 80;
            s1.sub2_marks = 90;
            s1.sub3_marks = 85;
            s1.getfullname();
            s1.getage();
            s1.totalmarks();

     Console.WriteLine("-------------------------------");

            Teacher t1 = new Teacher();
            t1.fname = "Hardik";
            t1.lname = "Molia";
            t1.age = 30;
            t1.salary = 50000;
            t1.Subject = "dot-net";

            t1.getfullname();
            t1.getage();

            Console.ReadLine();



        }
    }
}
```

OUTPUT:

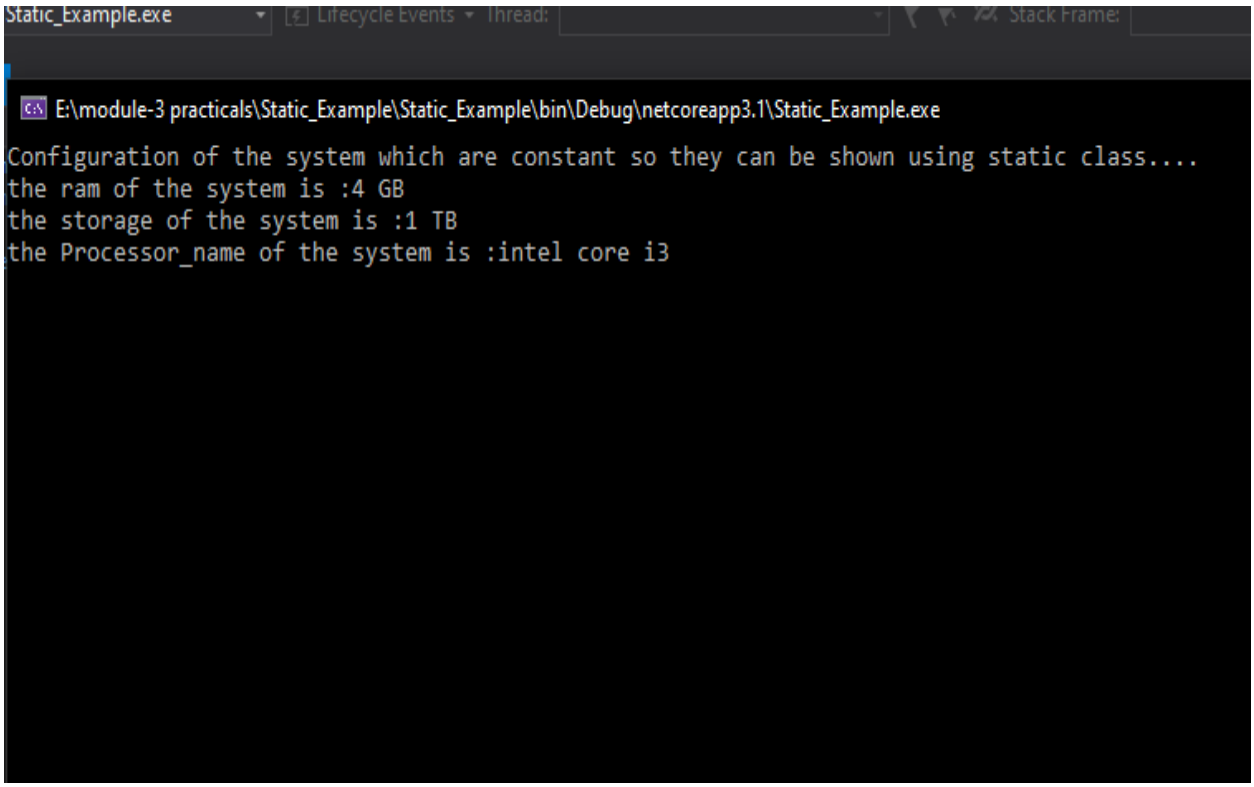


2. Static class Example.

```
using System;
using System.Runtime.CompilerServices;

namespace Static_Example
{
   static class SystemInfo
   {
      public static String storage;
      public static String Processor_name;
      public static String Ram;

      public static void getinfo()
      {
         Console.WriteLine("the ram of the system is :{0} ", Ram);
         Console.WriteLine("the storage of the system is :{0} ", storage);
         Console.WriteLine("the Processor_name of the system is :{0} ",
Processor_name);
      }
```

```
        static SystemInfo()
        {
            storage = "1 TB";
            Processor_name = "intel core i3";
            Ram = "4 GB";
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Configuration of the system which are constant so
theycan be shown using static class....");
     SystemInfo.getinfo();
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

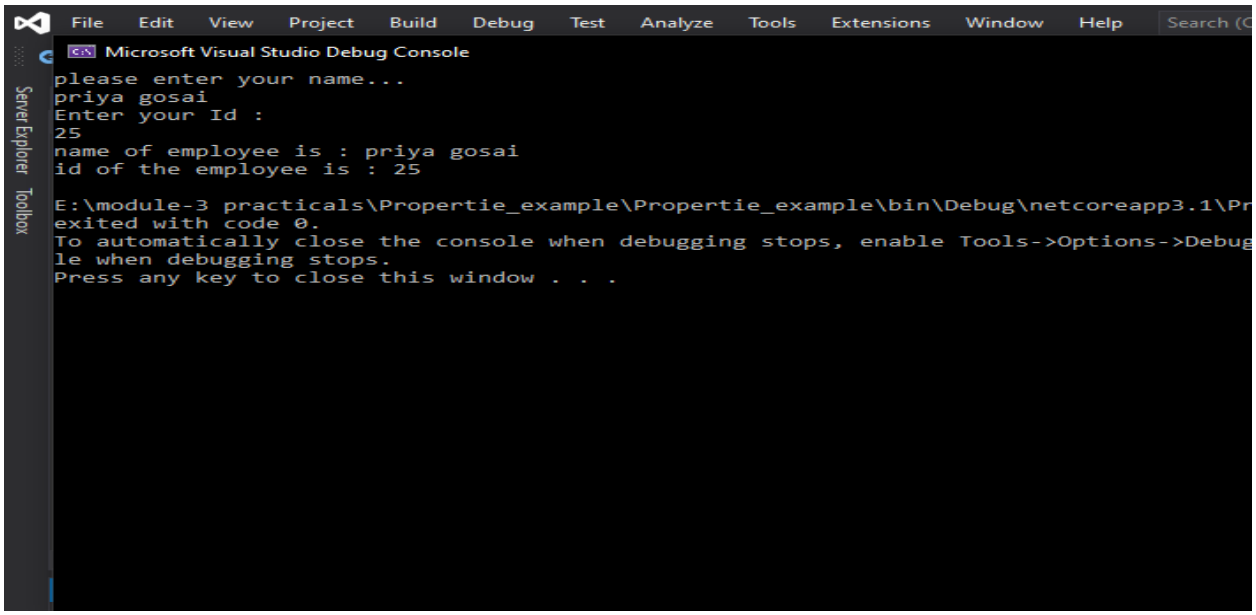


3.properties of class example

```
using System;

namespace Propertie_example
{
    class employee
    {
        private String _empname;
        private int _id;
        public int Id
        {
            set
            {
                if (value >= 1)
                {
                    this._id = value;
                }
                else
                {
                    Console.WriteLine("Invalid entry for id.....");
                }

            }
            get
            {
                return this._id;

            }
        }

        public String empname
        {
            set
            {
                if (String.IsNullOrEmpty(value))
                {
                    Console.WriteLine("name is required....");

                }
                else
                { this._empname = value; }

            }
            get
```

```
        {
            return this._empname;
        }

    }

}
class Program
{
    static void Main(string[] args)
    {
        employee e1 = new employee();
        Console.WriteLine("please enter your name...");
         String name = Console.ReadLine();

        e1.empname = name;
        Console.WriteLine("Enter your Id : ");
        int id = Convert.ToInt32((Console.ReadLine()));
        e1.Id = id;
        Console.WriteLine("name of employee is : {0}",e1.empname);
        Console.WriteLine("id of the employee is : {0}",e1.Id);

    }
}
}
```

OUTPUT:

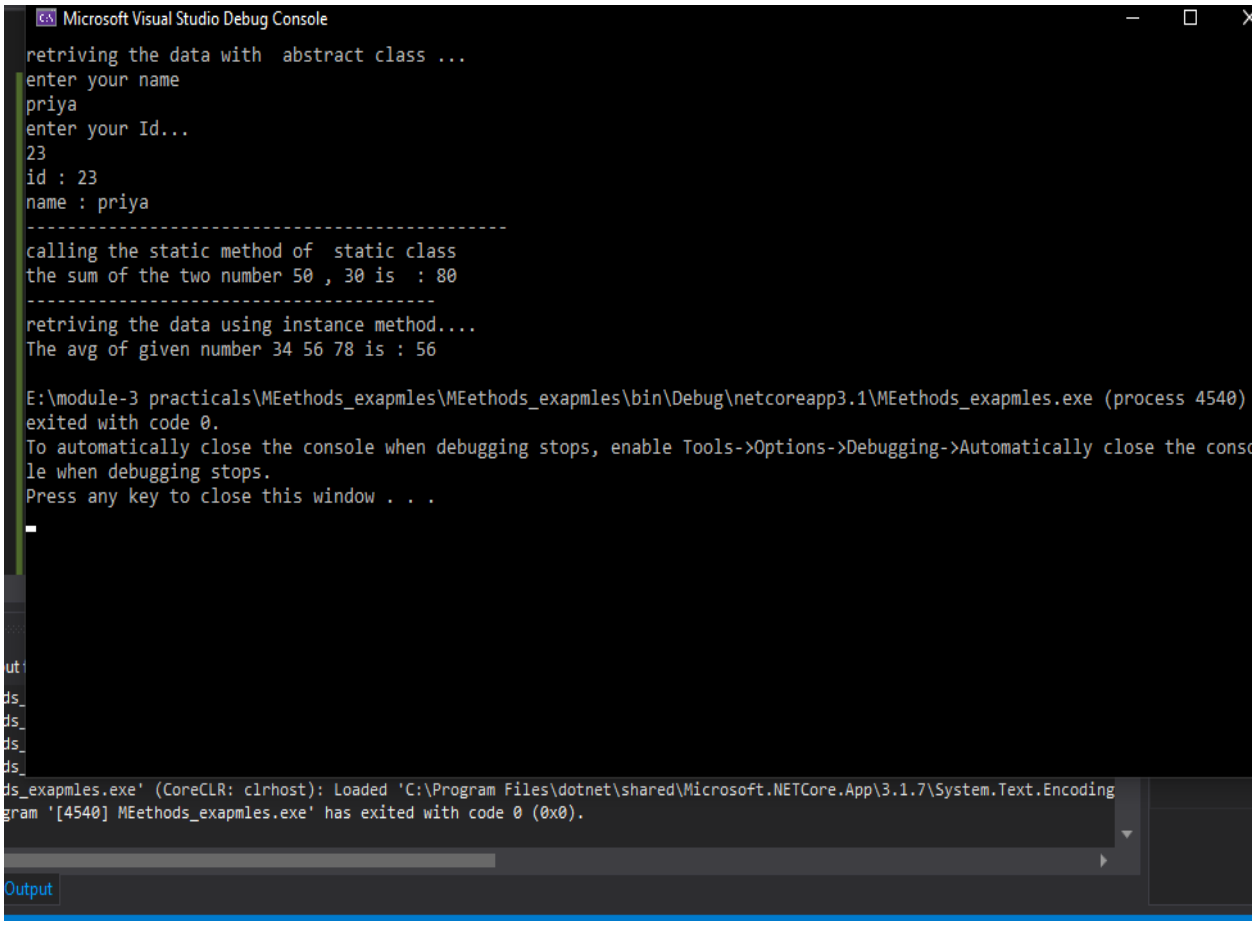


4.methods in c# example.

```
using System;
using System.Threading;
using System.Transactions;

namespace MEethods_exapmles
{
    abstract class test
    {
        public int Id ;
        public String name;
        public abstract  void getInfo();
    }
    class classA : test
    {
        public override  void getInfo()
        {
            Console.WriteLine("id : "+Id);
            Console.WriteLine("name : " + name);

        }

    }
    public static class test2
    {
        static int a ;
        static int b ;
        public static void sum(int a, int b)
        {
            int result = a + b;
            Console.WriteLine("the sum of the two number {0} , {1} is  : {2}",a,b,result);
        }

    }
    public class test3
    {
        public void getavg(int a, int b, int c)
        {
            int result = (a + b + c) / 3;
            Console.WriteLine("The avg of given number {0} {1} {2} is : {3}",a,b,c,result);
        }

    }
```

```
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("retriving the data with  abstract class ... ");
            Console.WriteLine("enter your name ");
            String nm = Console.ReadLine();
            Console.WriteLine("enter your Id... ");
            int ID = Convert.ToInt32(Console.ReadLine());

            classA a1 = new classA();
            a1.Id = ID;
            a1.name = nm;

            a1.getInfo();
            Console.WriteLine("-----------------------------------------------");
            Console.WriteLine("calling the static method of  static class");
            test2.sum(50,30);
            Console.WriteLine("---------------------------------------");
            Console.WriteLine("retriving the data using instance method....");
            test3 t3 = new test3();

            t3.getavg(34, 56, 78);
        }
    }
}
```

Output:

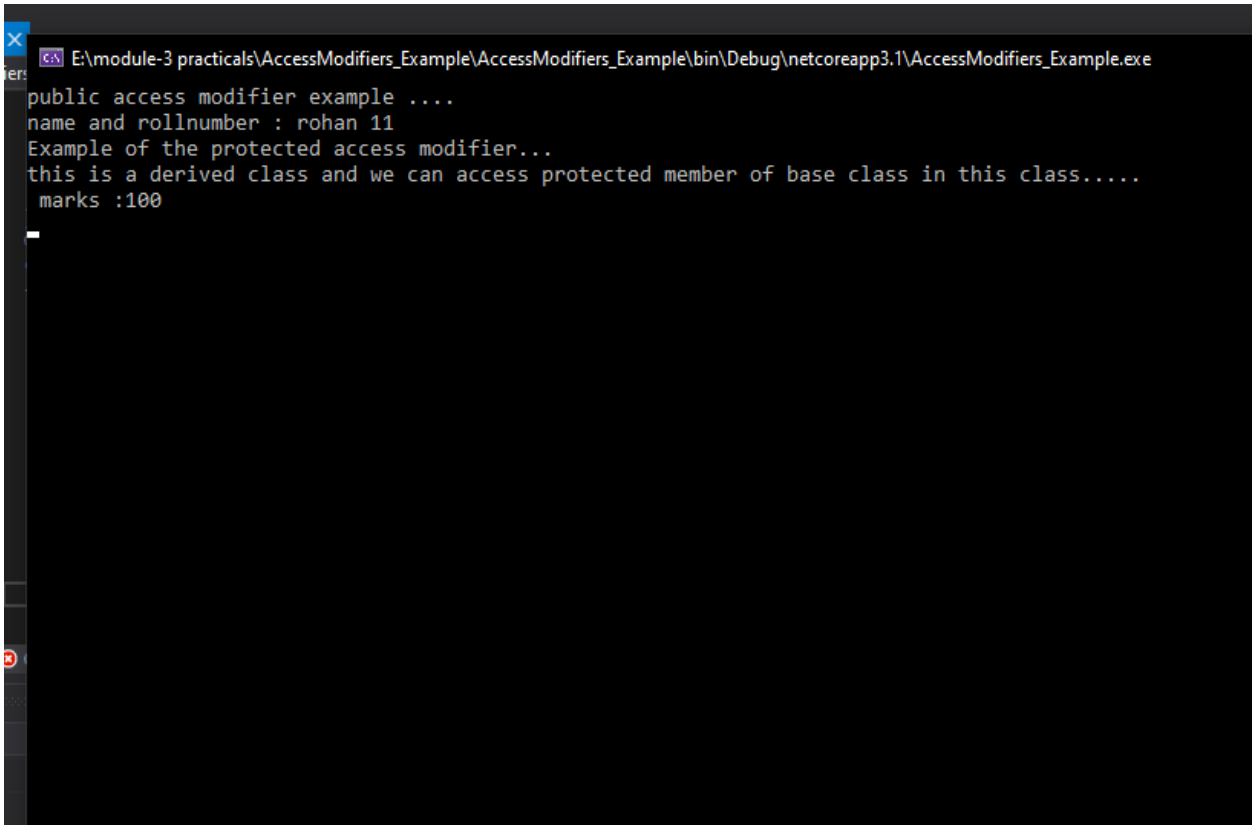


5.access modifiers example.

```
using System;

namespace AccessModifiers_Example
{
    class student
    {
        public int rollNo = 11;
        public String name = "rohan";
        private int mobileno;
        protected int marks = 100;


    }

    class test : student
    {

        public void getmarks()
        {
```

```
            Console.WriteLine("this is a derived class and we can access protected member
of base class in this class.....");
            Console.WriteLine(" marks :{0} ", marks);
        }

    }
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("public access modifier example ....");
            student s1 = new student();

            Console.WriteLine("name and rollnumber : {0} {1}", s1.name, s1.rollNo);
            Console.WriteLine("Example of the protected access modifier...");
            test t1 = new test();
            t1.getmarks();

            Console.ReadLine();

        }
    }
}
```

OUTPUT:



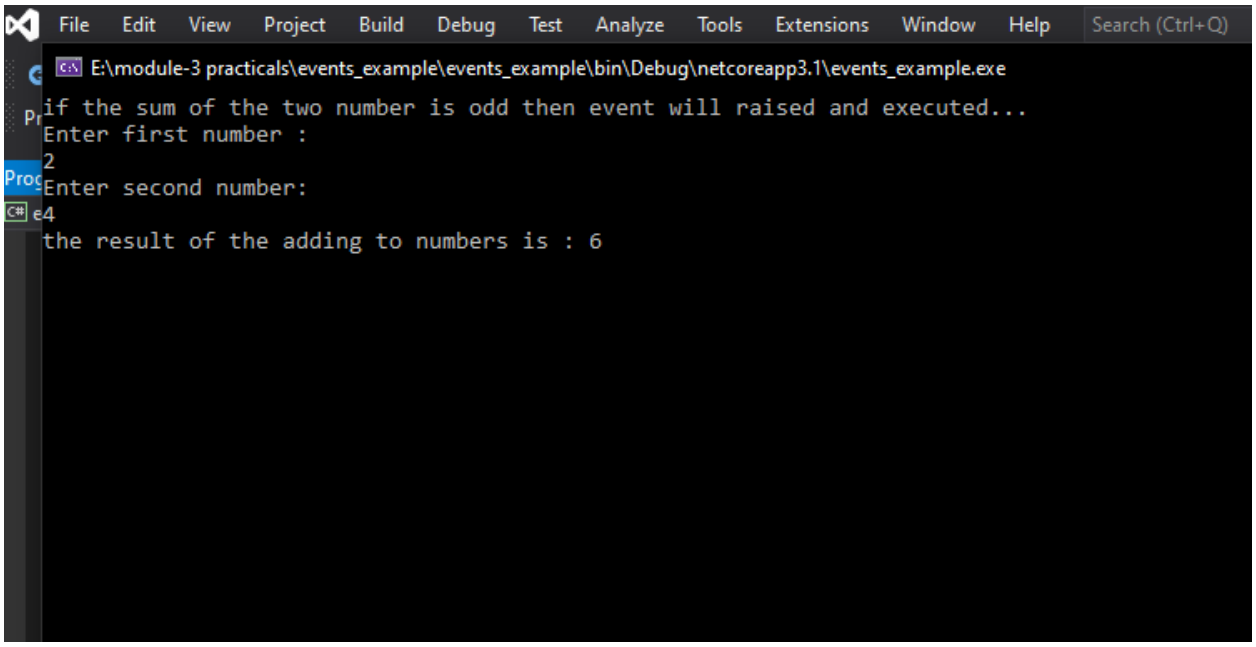
6.events example

```
using System;

namespace events_example
{
    class test
    {
        public delegate void oddnumbers();//decleare a delegate
        public event oddnumbers ev_oddnumbers;
        public void add(Int32 a, Int32 b)
        {
            Int32 result;
            result = a + b;
            Console.WriteLine("the result of the adding to numbers is : {0}", result);

            if ((result % 2 != 0) && (ev_oddnumbers != null))
            {
                ev_oddnumbers();//raised event
            }

        }
    }
    class Program
    {
        static void Eventmessage()
        {
            Console.WriteLine("event occured and handled : the sum of the given number
is a odd number ");

        }
        static void Main(string[] args)
        {
            Console.WriteLine("if the sum of the two number is odd then event will raised
and executed...");
            test t = new test();
            t.ev_oddnumbers += new test.oddnumbers(Eventmessage);
            Console.WriteLine("Enter first number : ");
            int  a = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter second number: ");
            int  b = Convert.ToInt32(Console.ReadLine());
            t.add(a,b);


            Console.ReadLine();

        }
```

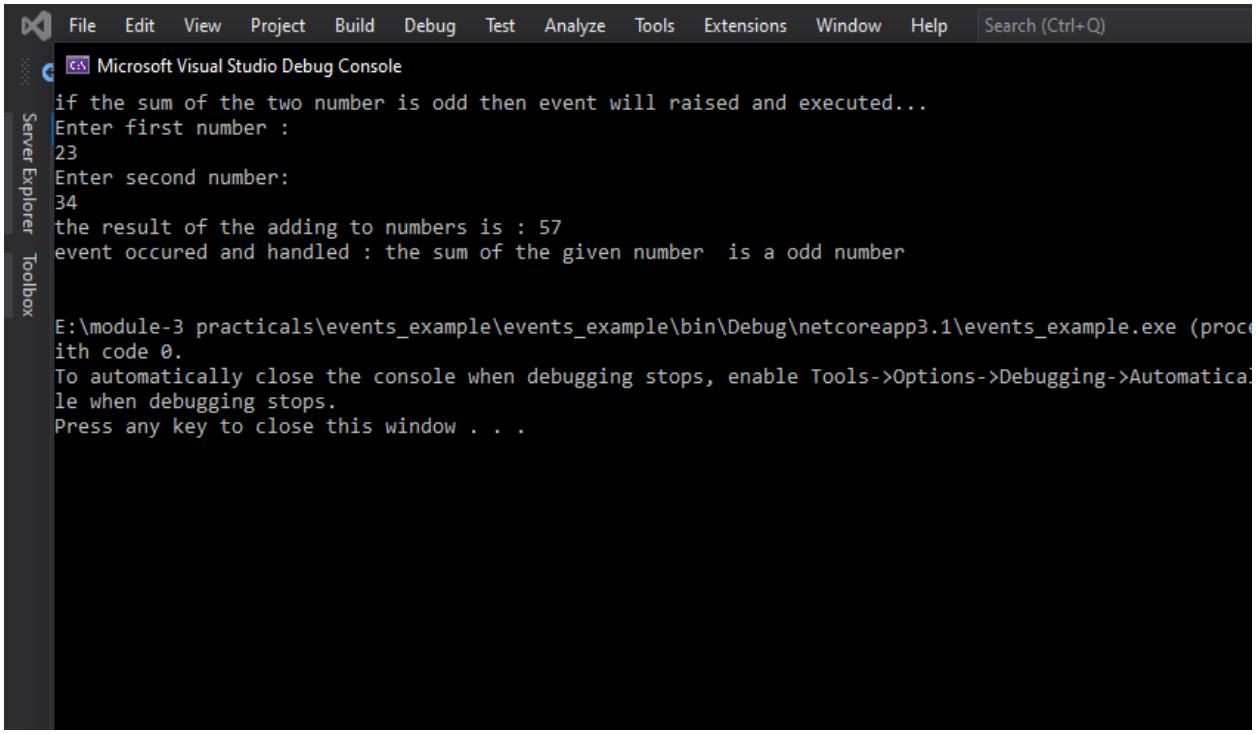
```
    }
}
```

OUTPUT:

If event will not occur than output will be as given:



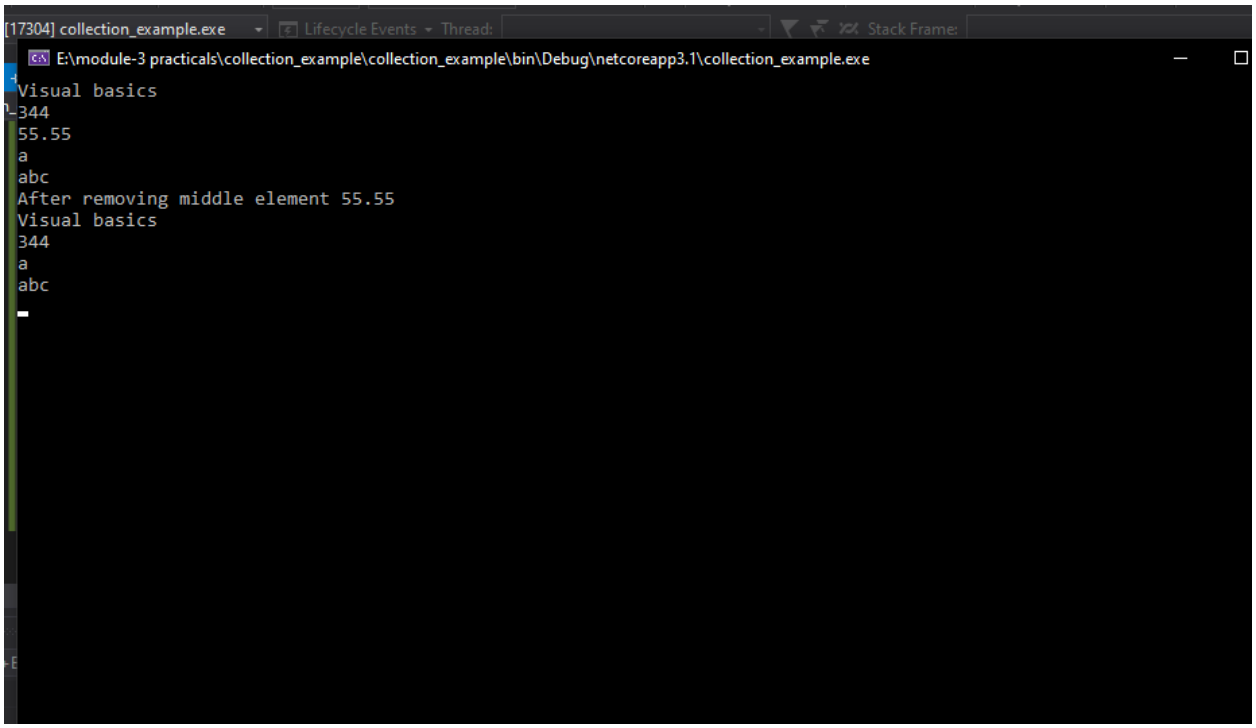
If event will occur than output will be as given:



7.collection example using array list.

```
using System;
using System.Collections;
using System.Linq.Expressions;

namespace collection_example
{
   class Program
   {
      static void Main(string[] args)
      {
         var data = new ArrayList();

         data.Add("Visual basics");
         data.Add(344);
         data.Add(55.55);
         data.Add('a');
         data.Add("abc");


         foreach (object el in data)
         {
            Console.WriteLine(el);
         }

         Console.WriteLine("After removing middle element 55.55 ");
         data.Remove(55.55);

         foreach (object el in data)
         {
            Console.WriteLine(el);
         }
         Console.ReadLine();
      }
   }
}
```

OUTPUT:


[17304] collection_example.exe
Lifecycle Events
Thread:
Stack Frame:
E:\module-3 practicals\collection_example\collection_example\bin\Debug\netcoreapp3.1\collection_example.exe
Visual basics
-344
55.55
a
abc
After removing middle element 55.55
Visual basics
344
a
abc