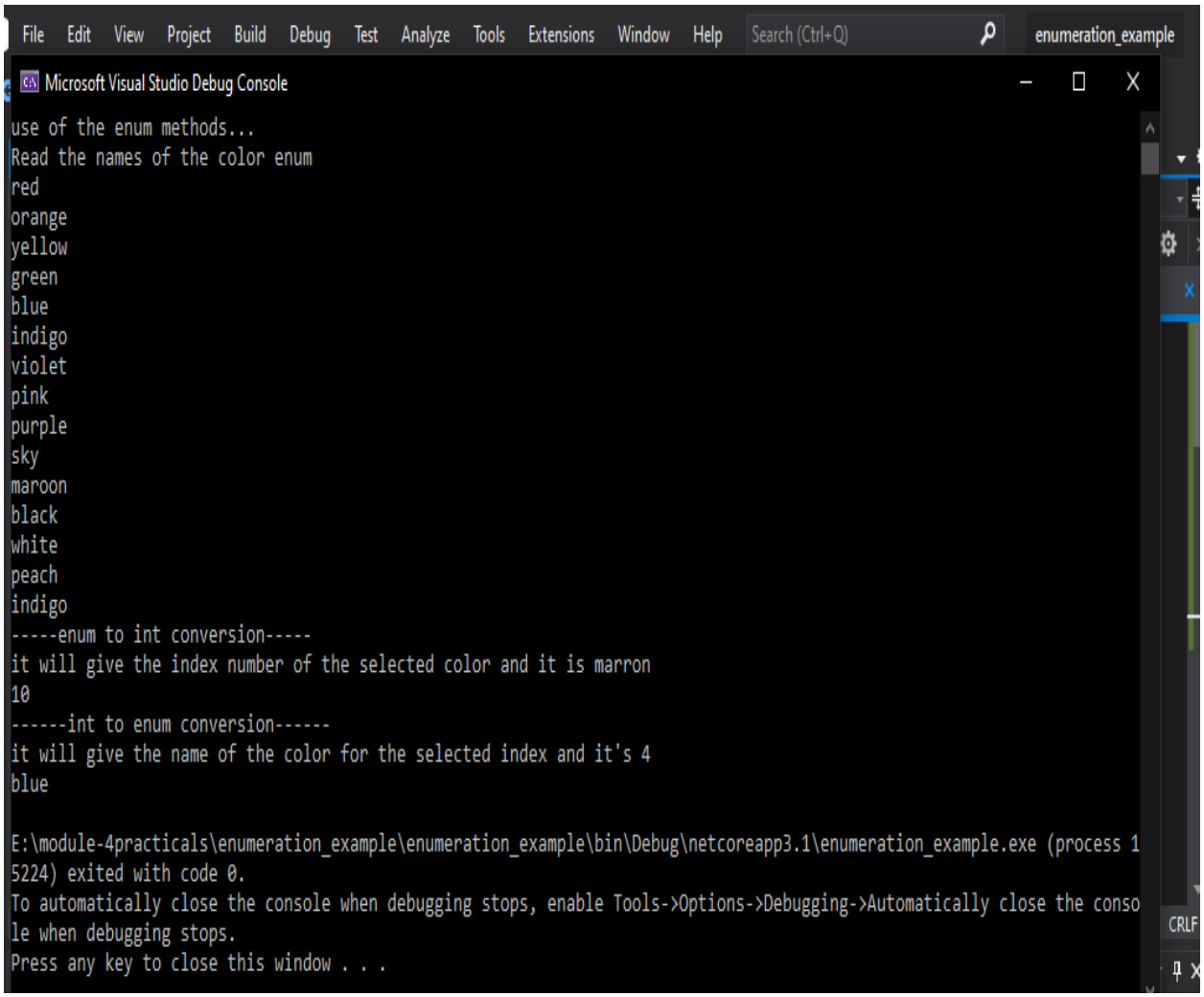
1.Example of enumeration

```
using System;

namespace enumeration_example
{
   public enum colors
      {
         red,
         orange,
         yellow,
         green,
         blue,
         indigo,
         violet,
         pink,
         purple,
         sky,
         maroon,
         black,
         white,
         peach,
      };

   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("use of the enum methods...");
         Console.WriteLine("Read the names of the color enum");
         foreach (string str in Enum.GetNames(typeof(colors)))
         {
            Console.WriteLine(str);
         }
         colors c = colors.indigo;
         Console.WriteLine(c);

         Console.WriteLine("-----enum to int conversion-----");
         Console.WriteLine("it will give the index number of the selected color and it is marron ");
         int color = (int)colors.maroon;

         Console.WriteLine(color);
         Console.WriteLine("------int to enum conversion------");
```

```
            Console.WriteLine("it will give the name of the color for the selected index and it's 4");
            var cl = (colors)4;
            Console.WriteLine(cl);
        }
    }
}
```

OUTPUT:

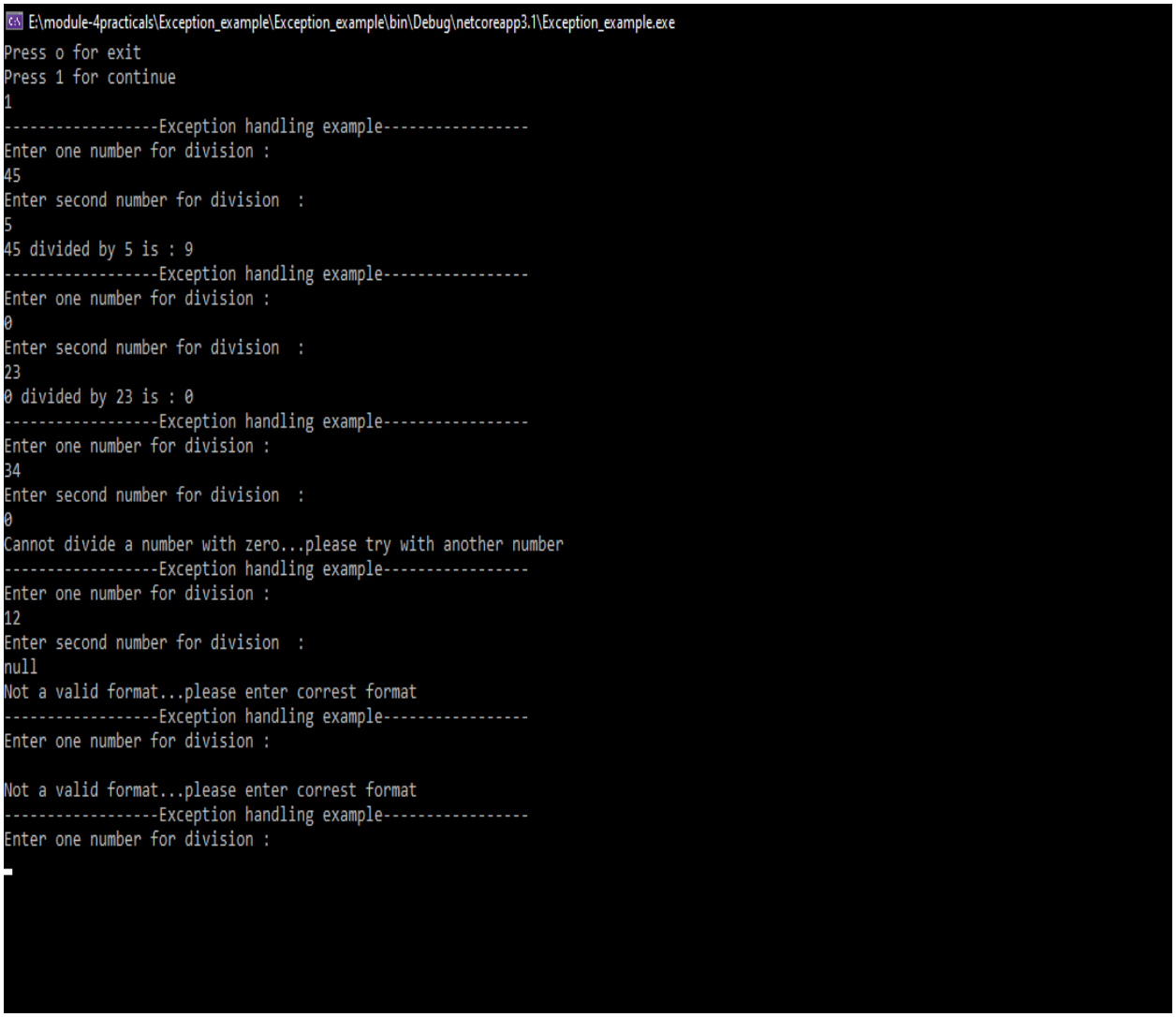


2.Example of Exception handling

```
using System;
using System.Xml.Serialization;

namespace Exception_example
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Press o for exit");
         Console.WriteLine("Press 1 for continue");
         int Choice = int.Parse(Console.ReadLine());

         while (Choice != 0)
         {s
            Console.WriteLine("------------------Exception handling example----------------");

            try
            {
               Console.WriteLine("Enter one number for division : ");
               int a = int.Parse(Console.ReadLine());
               Console.WriteLine("Enter second number for division  : ");
               int b = int.Parse(Console.ReadLine());
               decimal result = a / b;
               Console.WriteLine("{0} divided by {1} is : {2}", a, b, result);
            }
            catch (System.FormatException)
            {
               Console.WriteLine("Not a valid format...please enter correst format");
            }
            catch (DivideByZeroException)
            {
               Console.WriteLine("Cannot divide a number with zero...please try with another
number");
            }

            catch (InvalidOperationException)
            {
               Console.WriteLine("Invalid OPeration...");
            }
            catch (NullReferenceException)
```

```
            {
                Console.WriteLine("value can not be null...please enter number");
            }
        }
    }
  }
}
```

OUTPUT:

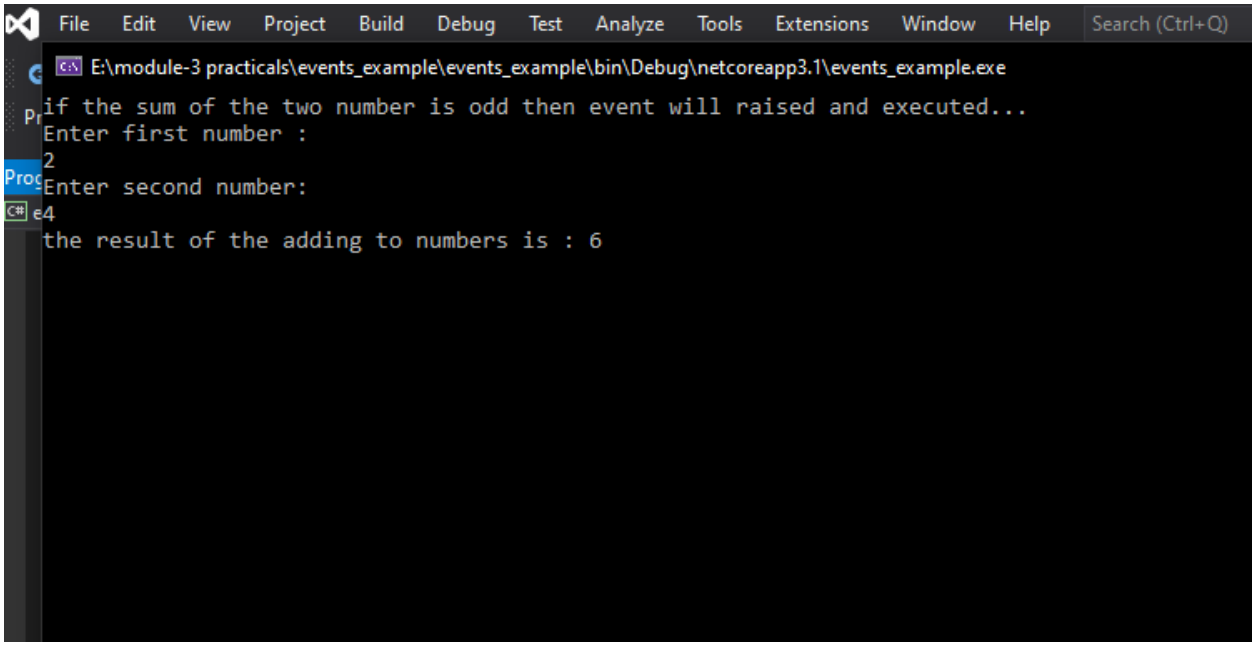


3.Example of the event

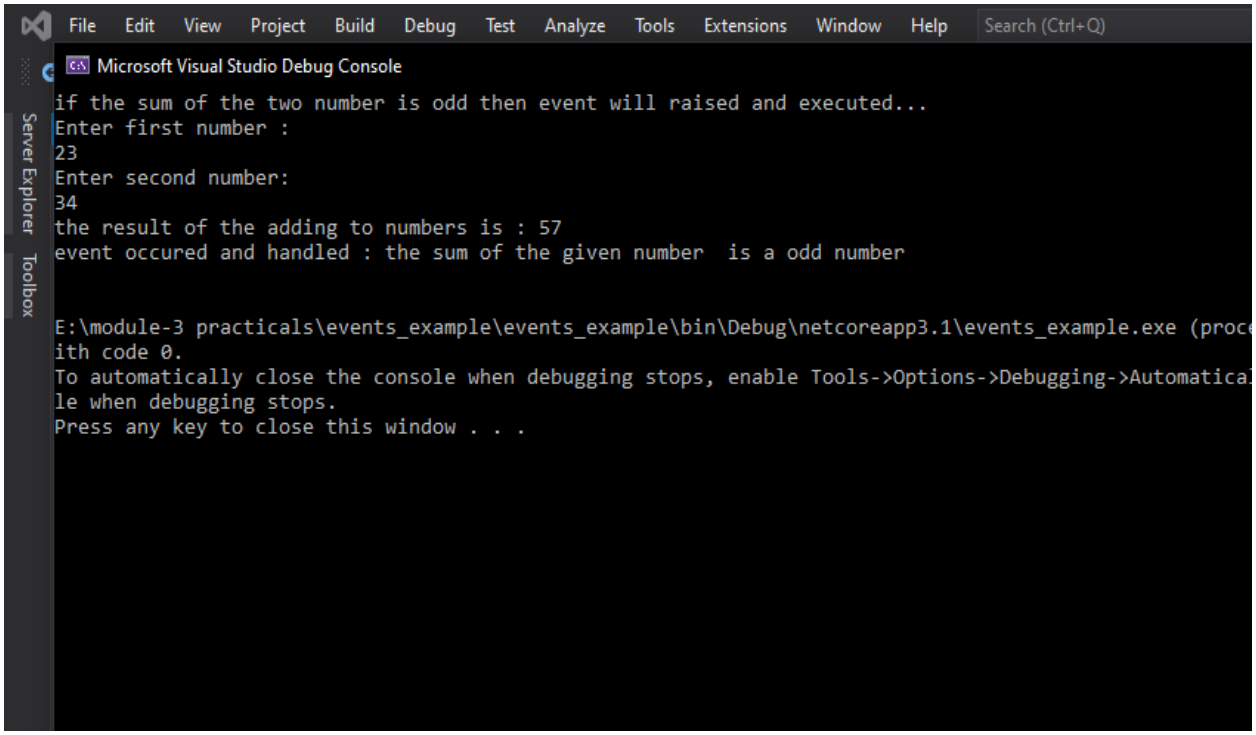
```
using System;
namespace events_example
{
    class test
    {
        public delegate void oddnumbers();//decleare a delegate
        public event oddnumbers ev_oddnumbers;
        public void add(Int32 a, Int32 b)
        {
            Int32 result;
            result = a + b;
            Console.WriteLine("the 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 the event will occur then te output will be as given:

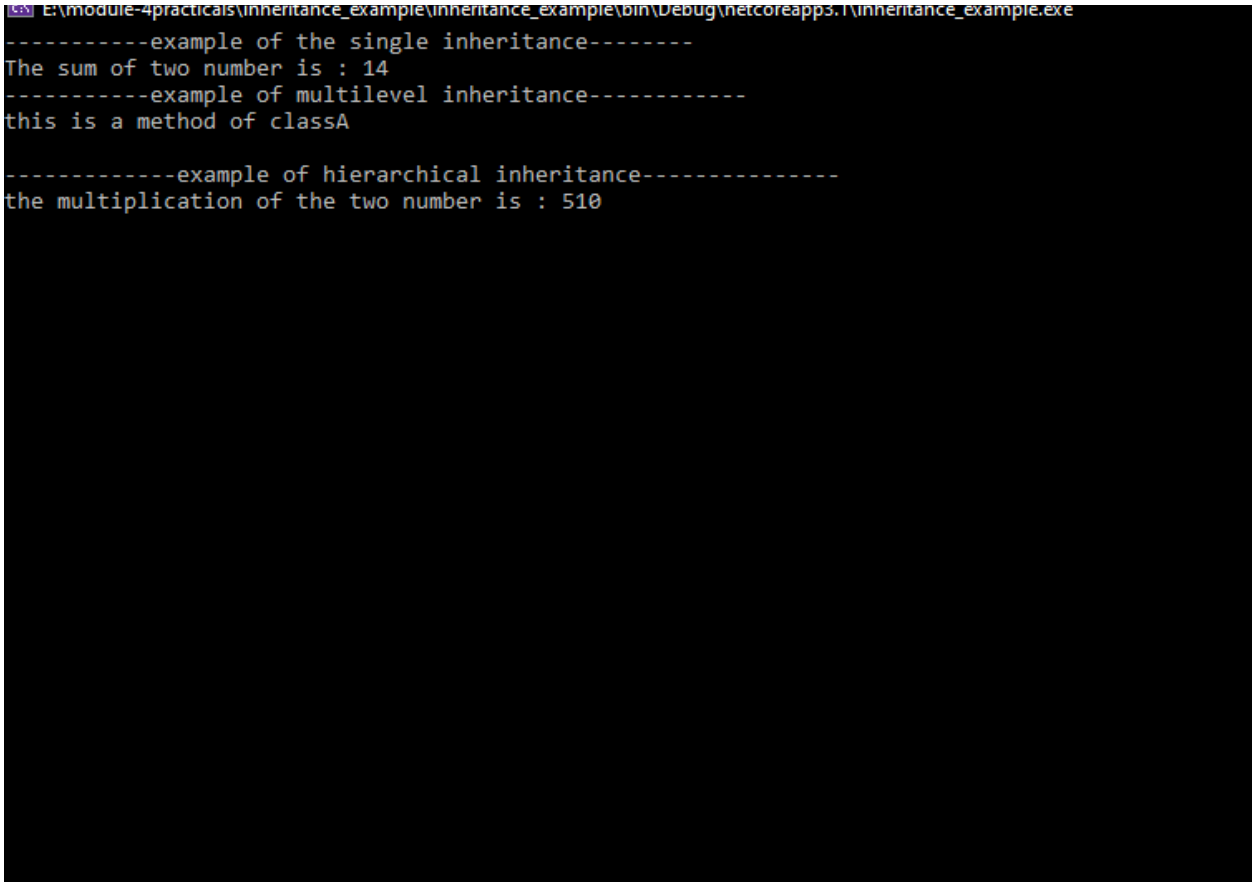


4.example of inheritance.

```
using System;

namespace inheritance_example
{
   class test
   {
      public int value1;
      public  int value2;
      public int seta(int a)
      {
         value1 = a;
         return (value1);
      }
      public int setb(int b)
      {
         value2 = b;
         return (value2);
      }
   }
   class classA : test // single inheritance
   {
      public int result;
      public int sum()
      {
         result = value1 + value2;
         Console.WriteLine("The sum of two number is : " +result);
         return (result);
      }
      public void show()
      {

         Console.WriteLine("this is a method of classA ");
      }

   }
   class classB : classA //multilevel inheritance
   {
      int avg;
      public void average()
      {
         avg = result / 2;
```

```
        Console.WriteLine("the average of the given numbers is : " +avg);
    }
  }
  class classC : test //hierarchical inheritance
  {
    int res;
    public void multiply()
    {
      res = value1 * value2;
      Console.WriteLine("the multiplication of the two number is : "+res);
    }
  }
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("-----------example of the single inheritance--------");
      classA a = new classA();
      a.seta(6);
      a.setb(8);
      a.sum();

      Console.WriteLine("-----------example of multilevel inheritance-----------");
      classB b = new classB();
      b.show();
      Console.WriteLine();
      Console.WriteLine("-------------example of hierarchical inheritance---------------");
      classC c = new classC();
      c.seta(15);
      c.setb(34);
      c.multiply();

      Console.ReadLine();

    }
  }
}
```

OUTPUT:

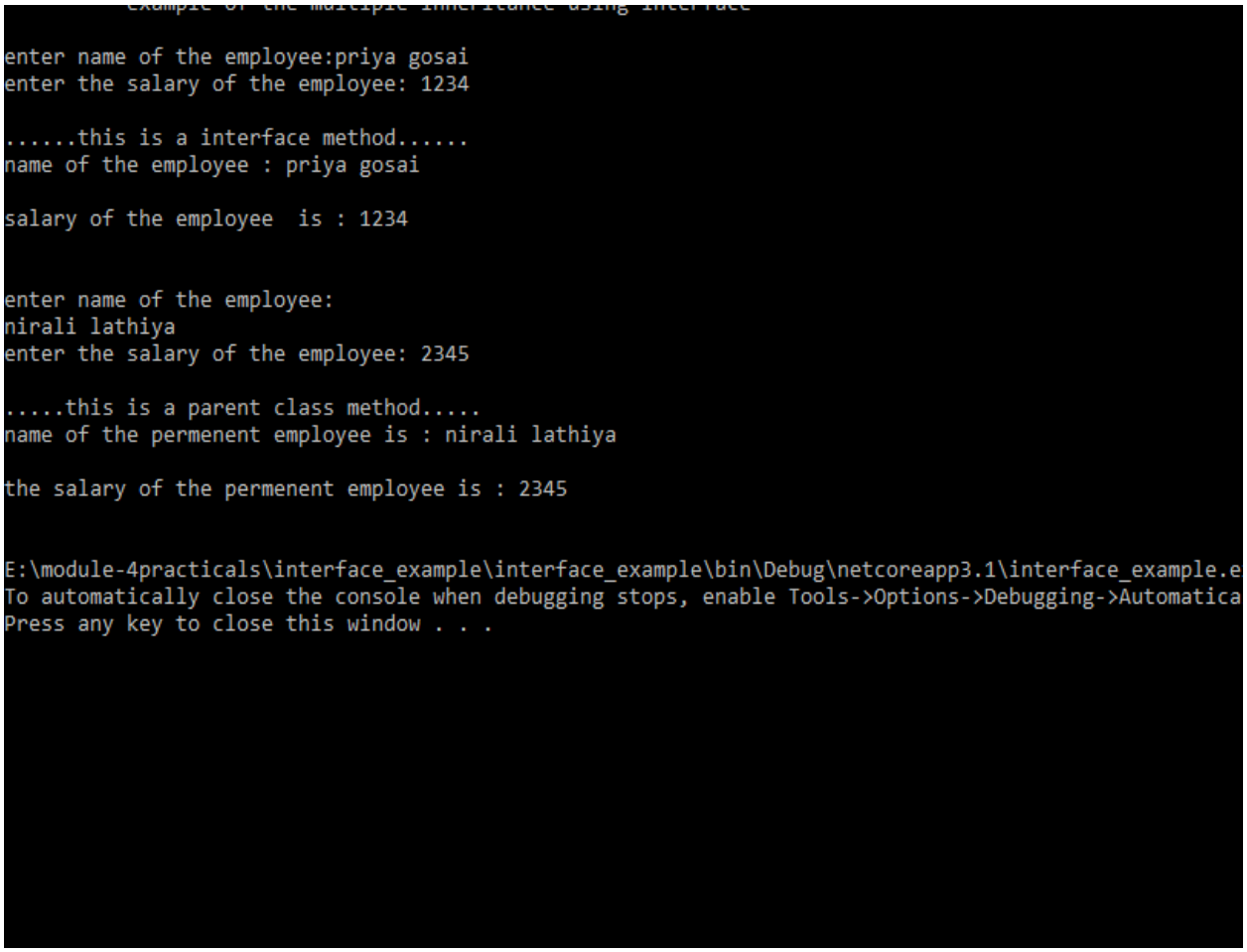

E:\module-4practicals\inheritance_example\inheritance_example\bin\Debug\netcoreapp3.1\inheritance_example.exe
----------example of the single inheritance--------
The sum of two number is : 14
----------example of multilevel inheritance-----------
this is a method of classA
-------------example of hierarchical inheritance---------------
the multiplication of the two number is : 510

5.Example of inface.

```
using System;

namespace interface_example
{
   interface iemployee
   {
       public void getname( string str);
      public void getsalary(uint slr);

   }
   class permenent_employee
   {
      public string ename;
      public uint salary;

      public void getename(string enm)
      {
         Console.WriteLine(".....this is a parent class method.....");
         ename = enm;
         Console.WriteLine("name of the permenent employee is : " +ename);
         Console.WriteLine();
      }
      public void getslr(uint eslry)
      {
         salary = eslry;
         Console.WriteLine("the salary of the permenent employee is : "+salary);
         Console.WriteLine();
      }

   }
   class salary :  permenent_employee , iemployee  //example of multiple inheritance
   {
      public void getname( string str)
      {
         Console.WriteLine("......this is a interface method......");
         string name = str;
         Console.WriteLine("name of the employee : " +name);
         Console.WriteLine();
      }
      public void getsalary(uint slr)
      {
```

```
        uint salary = slr;
        Console.WriteLine("salary of the employee  is : {0}",salary);
        Console.WriteLine();
    }

  }
  class Program
  {
    static void Main(string[] args)
    {
        Console.WriteLine("----------example of the multiple inheritance using interface-------");

        Console.WriteLine();
        Console.Write("enter name of the employee:");

        string str = Console.ReadLine();
        Console.Write("enter the salary of the employee: ");
        uint slr = uint.Parse(Console.ReadLine());
        Console.WriteLine();
        salary s1 = new salary();
        s1.getname(str);
        s1.getsalary(slr);


        Console.WriteLine();
        Console.Write("enter name of the employee:");
        Console.WriteLine();
        string enm = Console.ReadLine();
        Console.Write("enter the salary of the employee: ");
        uint eslry = uint.Parse(Console.ReadLine());
        Console.WriteLine();
        s1.getename(enm);
        s1.getslr(eslry);


    }
  }
}
```

OUTPUT:


enter name of the employee:priya gosai
enter the salary of the employee: 1234
......this is a interface method......
name of the employee : priya gosai
salary of the employee is : 1234
enter name of the employee:
nirali lathiya
enter the salary of the employee: 2345
.....this is a parent class method.....
name of the permenent employee is : nirali lathiya
the salary of the permenent employee is : 2345
E:\module-4practicals\interface_example\interface_example\bin\Debug\netcoreapp3.1\interface_example.e
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatica
Press any key to close this window . . .

6.Example of file handling cfreating a file.

```
using System;
using System.IO;


namespace filedemo
{
   class Program
   {
      static void Main(string[] args)
      {
         string filepath = @"E:\MyFile.txt";
         using (FileStream fs = new FileStream(filepath, FileMode.OpenOrCreate)) ;
         {
            if (File.Exists(filepath))
            {
               Console.WriteLine("File Created Successfully.....");
            }
            Console.ReadLine();          }
      }
   }
}
```

7. Read write and Append file

using System;

```
using System.IO;

namespace FileAppendDemo

{

   class Program

   {

      static void Main(string[] args)

      {

         string filepath = @"E:\MyFile.txt";

using (FileStream fs = new FileStream(filepath, FileMode.OpenOrCreate,
FileAccess.ReadWrite))

         {

            try

            {

               using (StreamWriter writer = new StreamWriter(fs))

               {

                  writer.WriteLine("This is MyFile");

                  writer.WriteLine("My name is Nirali");

               }

            }

            catch (Exception e)

            {

               Console.WriteLine(e.Message);

            }

         }
```

```
        using (FileStream fs = new FileStream(filepath, FileMode.OpenOrCreate,
FileAccess.ReadWrite))
      {
        try
        {
          using (StreamWriter writer = new StreamWriter(fs))
          {
            writer.WriteLine("This is MyFile");
            writer.WriteLine("My name is Nirali");
          }


        }
        catch (Exception e)
        {
          Console.WriteLine(e.Message);
        }

        using (FileStream fs1 = new FileStream(filepath, FileMode.Append,
FileAccess.Write))
        {
          try
          {
            using (StreamWriter writer = new StreamWriter(fs1))
            {
```

```
            writer.WriteLine("This is append operation");


        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
}


try
{


    using (StreamReader reader = new StreamReader(filepath))
    {
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            Console.WriteLine(line);
        }
    }
}
catch (Exception e)
{
```

```
            Console.WriteLine(e.Message);
        }

        Console.ReadLine();
    }
  }
 }
}
```