**Assignment (Sunday) - Rahul Kumar**

1. Exception Handling

using System;

namespace ExceptionHandling

{

    class DivideProg

    {

        int result;

        DivideProg()

        {

            result = 0;

        }

        public void division(int num1, int num2)

        {

            try {

                result = num1 / num2;

            }

            catch (DivideByZeroException exc)

            {

                Console.WriteLine("Exception caught: {0}", exc);

            }

            finally

            {

                Console.WriteLine("Result: {0}", result);

            }

        }

        static void Main(string[] args) {

            DivideProg div = new DivideProg();

            div.division(12, 0);

        }

    }

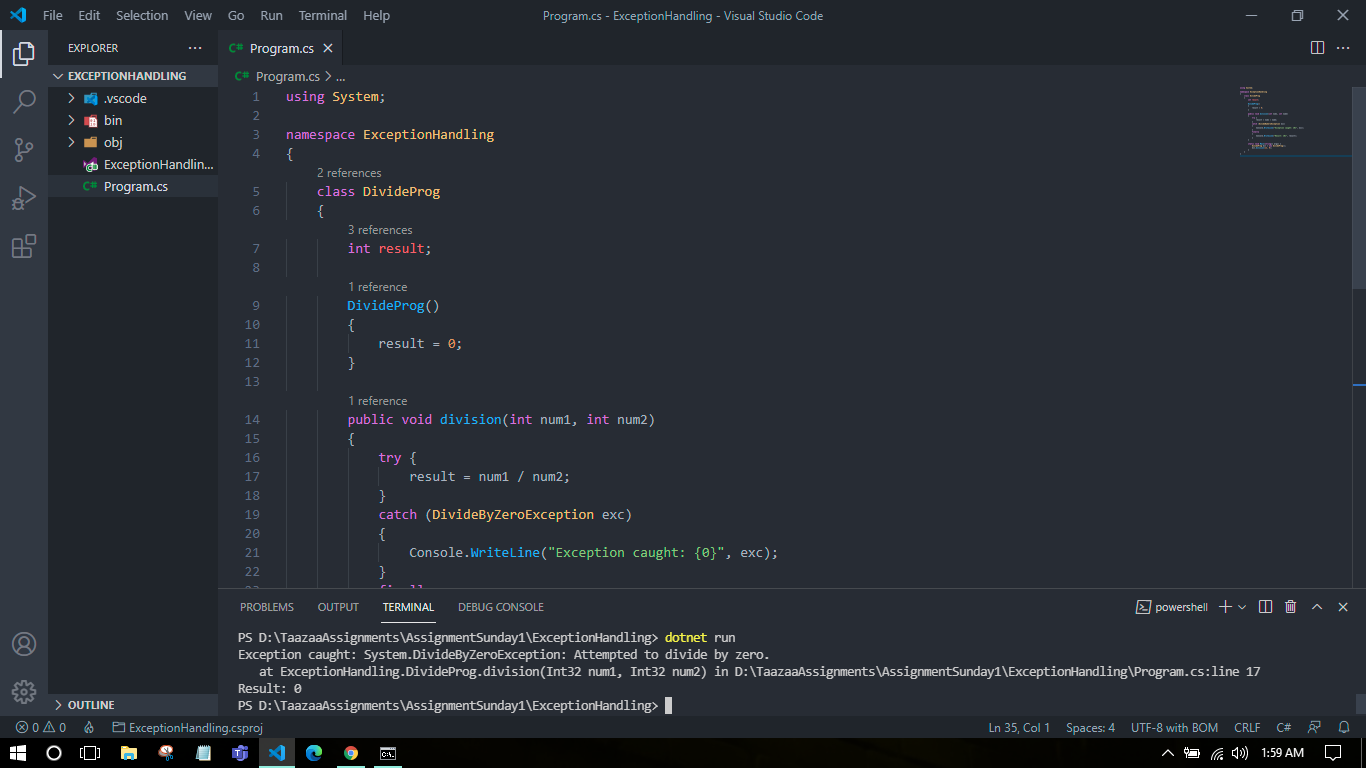
}

PS D:\TaazaaAssignments\AssignmentSunday1\ExceptionHandling> dotnet run

Exception caught: System.DivideByZeroException: Attempted to divide by zero.

at ExceptionHandling.DivideProg.division(Int32 num1, Int32 num2) in D:\TaazaaAssignments\AssignmentSunday1\ExceptionHandling\Program.cs:line 17

Result: 0



1. Func <>

Func is built-in delegate type and it must return a value.

using System;

namespace FuncDelegate

{

    class Program

    {

        public static int SumOfNum(int x, int y)

        {

            return x + y;

        }

        public static void Main()

        {

            Func<int,int, int> add = SumOfNum;

            int result = add(10, 20);

            Console.WriteLine(result);

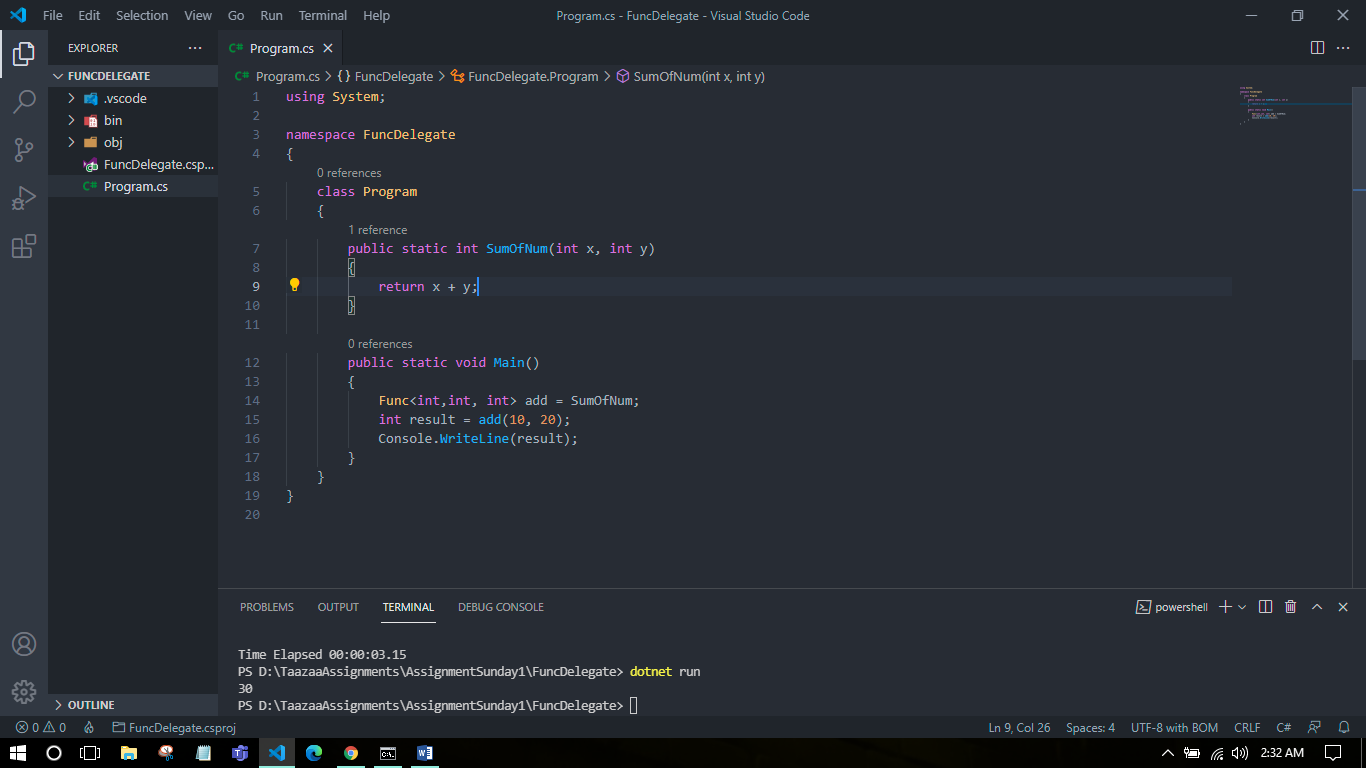
        }

    }

}

PS D:\TaazaaAssignments\AssignmentSunday1\FuncDelegate> dotnet run

30



1. Action <>

Action delegate is same as func delegate except that it does not return anything. Return type must be void.

using System;

namespace ActionDelegate

{

    class Program

    {

        public static void Main()

        {

            Action<int> ActionDel = ConsolePrint;

            ActionDel(25);

        }

        public static void ConsolePrint(int i)

        {

            Console.WriteLine(i);

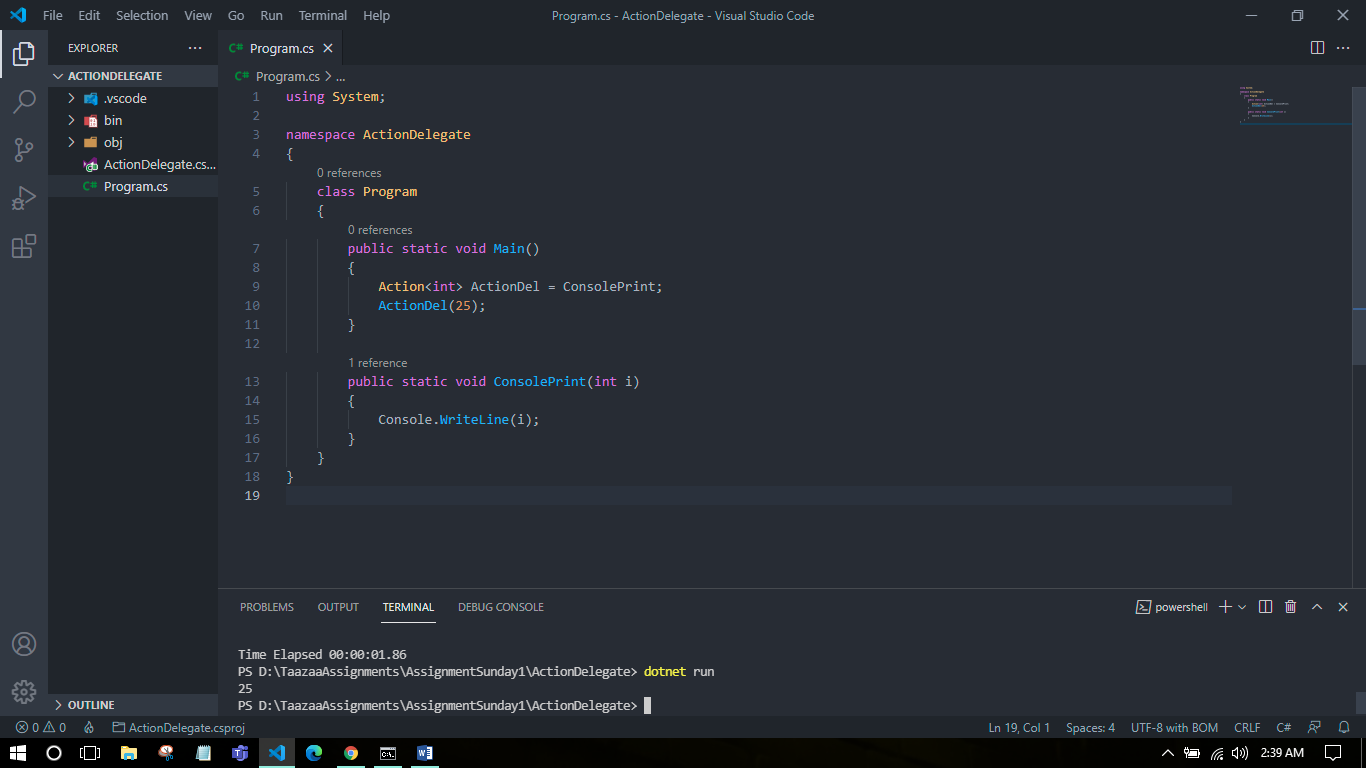
        }

    }

}

PS D:\TaazaaAssignments\AssignmentSunday1\ActionDelegate> dotnet run

25



1. Predicate <>

Predicate delegate takes one input parameter and boolean return type.

using System;

namespace PredicateDelegate

{

    class Program

    {

        public static void Main()

        {

            Predicate<string> isUpper = IsUpperCase;

            bool result = isUpper("hello world!!");

            Console.WriteLine(result);

        }

        public static bool IsUpperCase(string str)

        {

            return str.Equals(str.ToUpper());

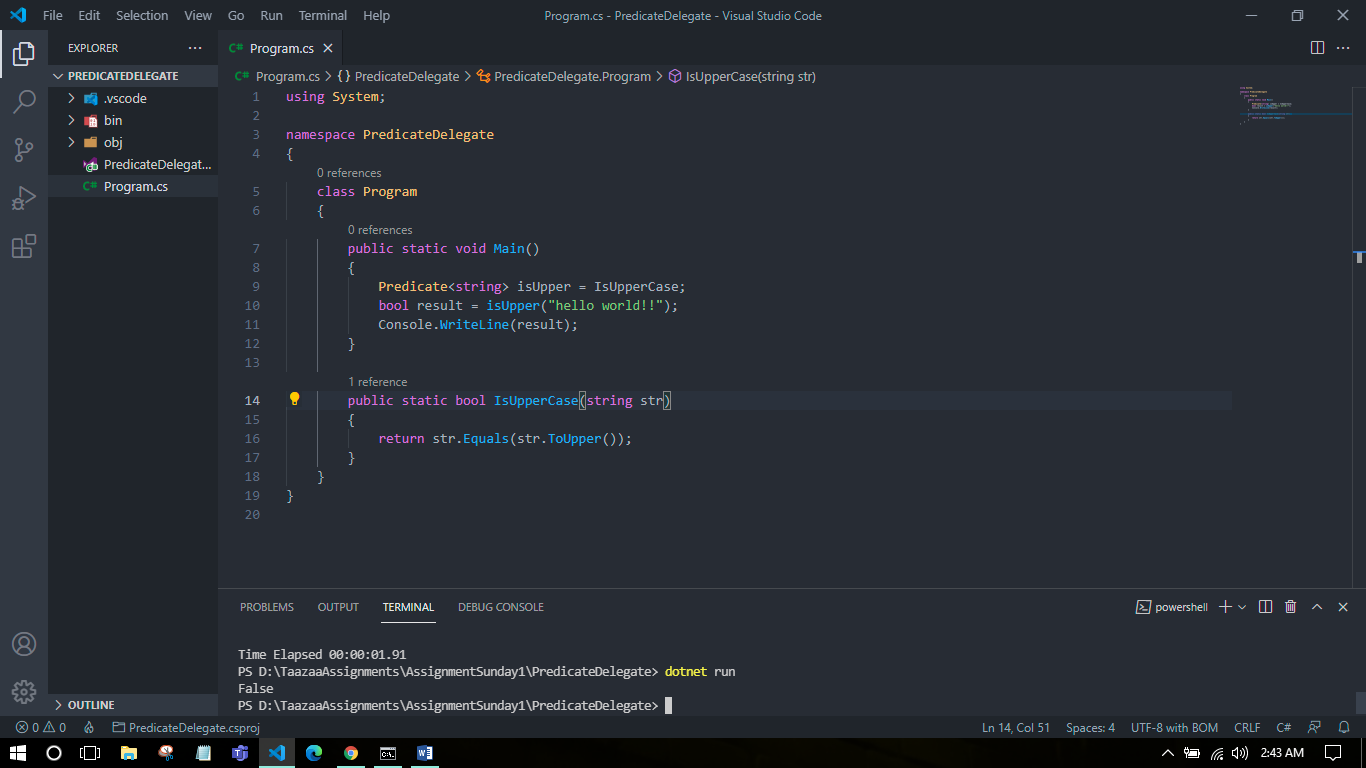
        }

    }

}

PS D:\TaazaaAssignments\AssignmentSunday1\PredicateDelegate> dotnet run

False



1. File Handling

When we open a file for reading or writing, it becomes stream. Stream is a sequence of bytes traveling from a source to a destination over a communication path.

The two basic streams are input and output streams. Input stream is used to read and output stream is used to write.

The System.IO namespace includes various classes for file handling.

using System;

using System.IO;

namespace FileHandling

{

    class Program

    {

        static void Main(string[] args)

        {

            FileStream ObjF = new FileStream("test.dat", FileMode.OpenOrCreate,

            FileAccess.ReadWrite);

            for (int i = 1; i <= 20; i++)

            {

            ObjF.WriteByte((byte)i);

            }

            ObjF.Position = 0;

            for (int i = 0; i <= 22; i++)

            {

                Console.Write(ObjF.ReadByte() + " ");

            }

            ObjF.Close();

        }

    }

}

PS D:\TaazaaAssignments\AssignmentSunday1\FileHandling> dotnet run

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 -1 -1 -1

