1. Delegate

using System;

namespace Delagate.Models

{

    public class DelegateAssign

    {

        public static double AddNum(int x, int y, int z)

        {

            return x+y+z;

        }

        public static void SubNum(int a, int b)

        {

            Console.WriteLine(a-b);

        }

        public static bool CheckLength(string str)

        {

            if(str.Length>5)

                return true;

            return false;

        }

    }

}

using System;

using Delagate.Models;

public delegate double DelegateAdd(int x, int y, int z);

public delegate void DelagateSub(int x, int y);

public delegate bool DelegateCheckLen(string st);

namespace Delagate

{

    class Program

    {

        static void Main(string[] args)

        {

            DelegateAdd addObj = new DelegateAdd(DelegateAssign.AddNum);

            double resAdd = addObj.Invoke(5, 34, 55);

            Console.WriteLine(resAdd);

            DelagateSub subObj = new DelagateSub(DelegateAssign.SubNum);

            subObj.Invoke(98, 56);

            DelegateCheckLen checkLen = new DelegateCheckLen(DelegateAssign.CheckLength);

            bool resLen = checkLen.Invoke("Hello There");

            Console.WriteLine(resLen);

        }

    }

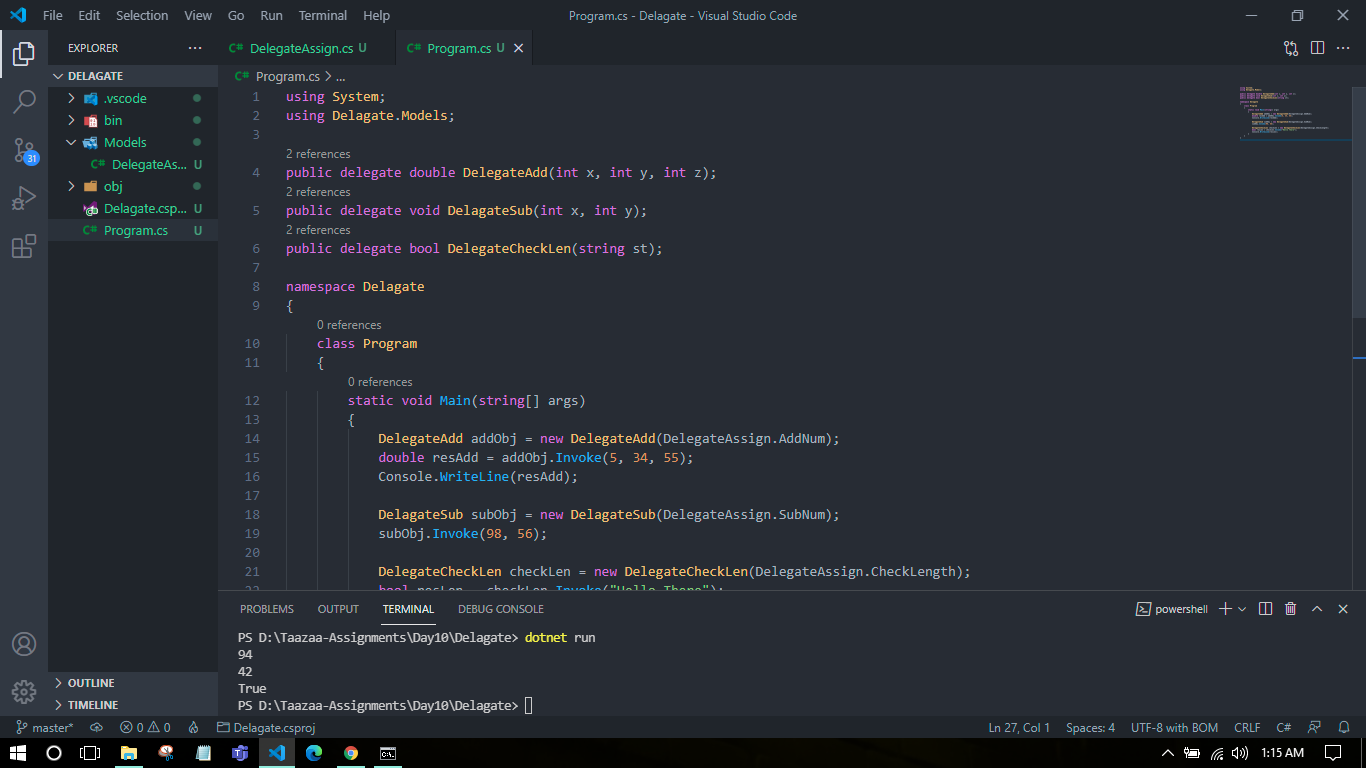
}

PS D:\Taazaa-Assignments\Day10\Delagate> dotnet run

94

42

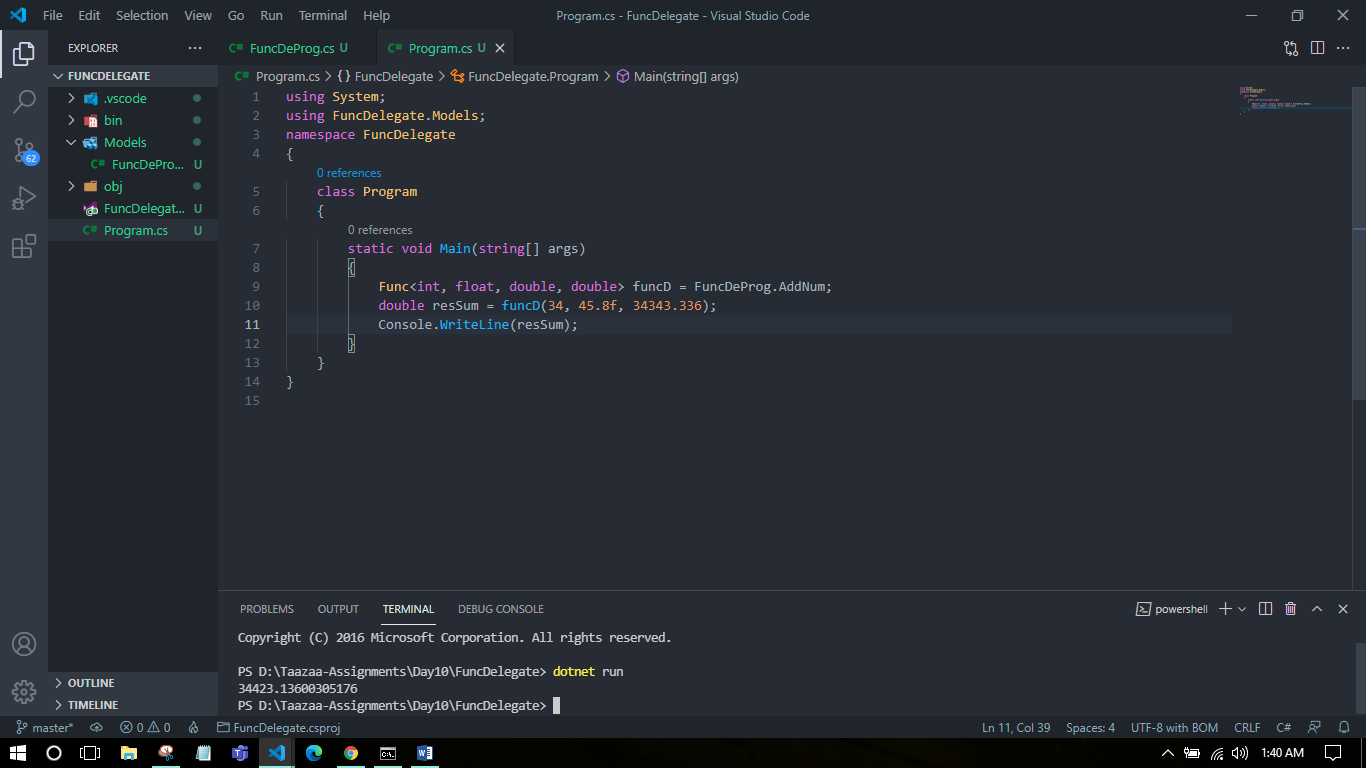
True



1. Func Delegate
2. namespace FuncDelegate.Models
3. {
4. public class FuncDeProg
5. {
6. public static double AddNum(int x, float y, double z)
7. {
8. return x+y+z;
9. }
10. }
11. }
12. using System;
13. using FuncDelegate.Models;
14. namespace FuncDelegate
15. {
16. class Program
17. {
18. static void Main(string[] args)
19. {
20. Func<int, float, double, double> funcD = FuncDeProg.AddNum;
21. double resSum = funcD(34, 45.8f, 34343.336);
22. Console.WriteLine(resSum);
23. }
24. }
25. }

PS D:\Taazaa-Assignments\Day10\FuncDelegate> dotnet run

34423.13600305176



3. Action Delegate

using System;

namespace ActionDelegate.Models

{

    public class ActionDelProg

    {

        public static void SubNum(int a, double b)

        {

            Console.WriteLine(a-b);

        }

    }

}

using System;

using ActionDelegate.Models;

namespace ActionDelegate

{

    class Program

    {

        static void Main(string[] args)

        {

            Action<int, double> actObj = ActionDelProg.SubNum;

            actObj.Invoke(132, 54.342);

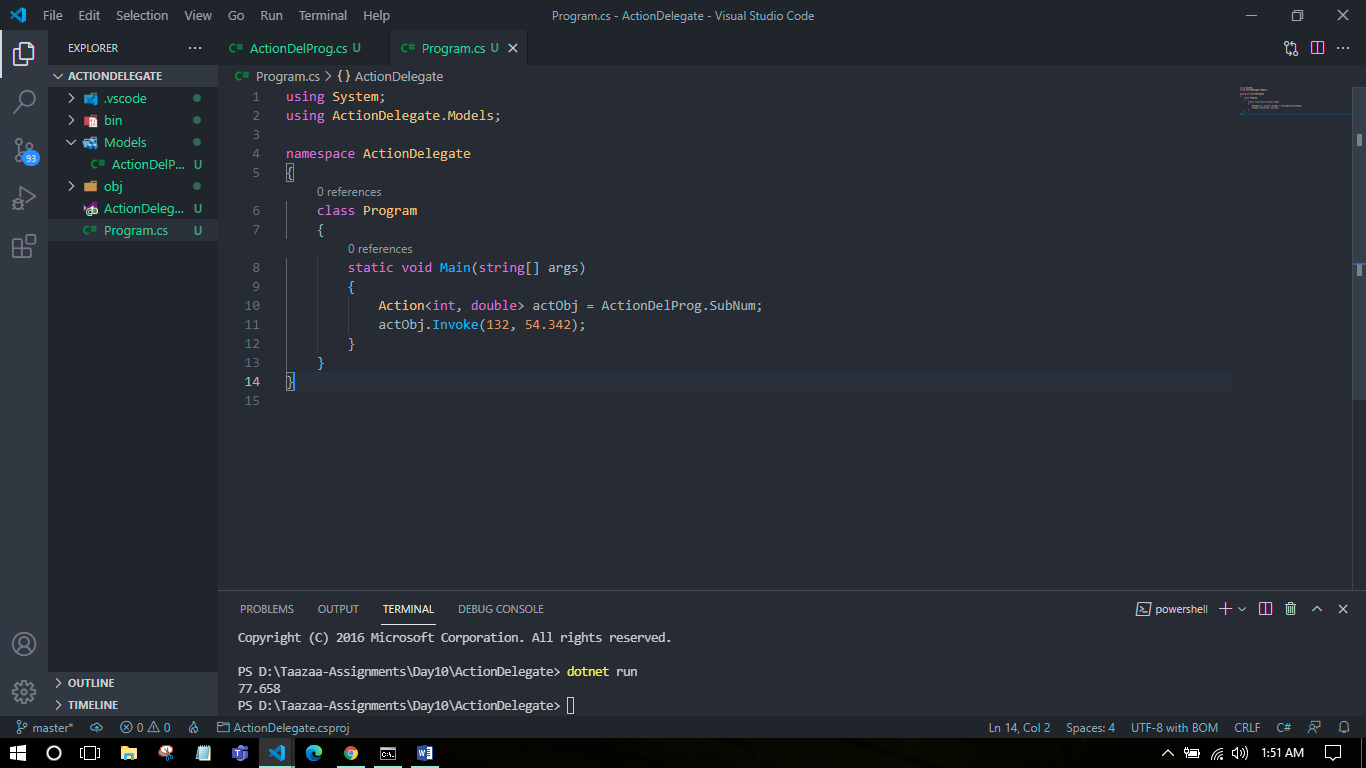
        }

    }

}

PS D:\Taazaa-Assignments\Day10\ActionDelegate> dotnet run

77.658



4. Predicate delegate

namespace PredicateDelegate.Models

{

    public class predicateDelProg

    {

        public static bool CheckLength(string str)

        {

            if(str.Length>8)

                return true;

            return false;

        }

    }

}

using System;

using PredicateDelegate.Models;

namespace PredicateDelegate

{

    class Program

    {

        static void Main(string[] args)

        {

            Predicate<string> preObj = predicateDelProg.CheckLength;

            bool resLen = preObj.Invoke("Hello There");

            Console.WriteLine(resLen);

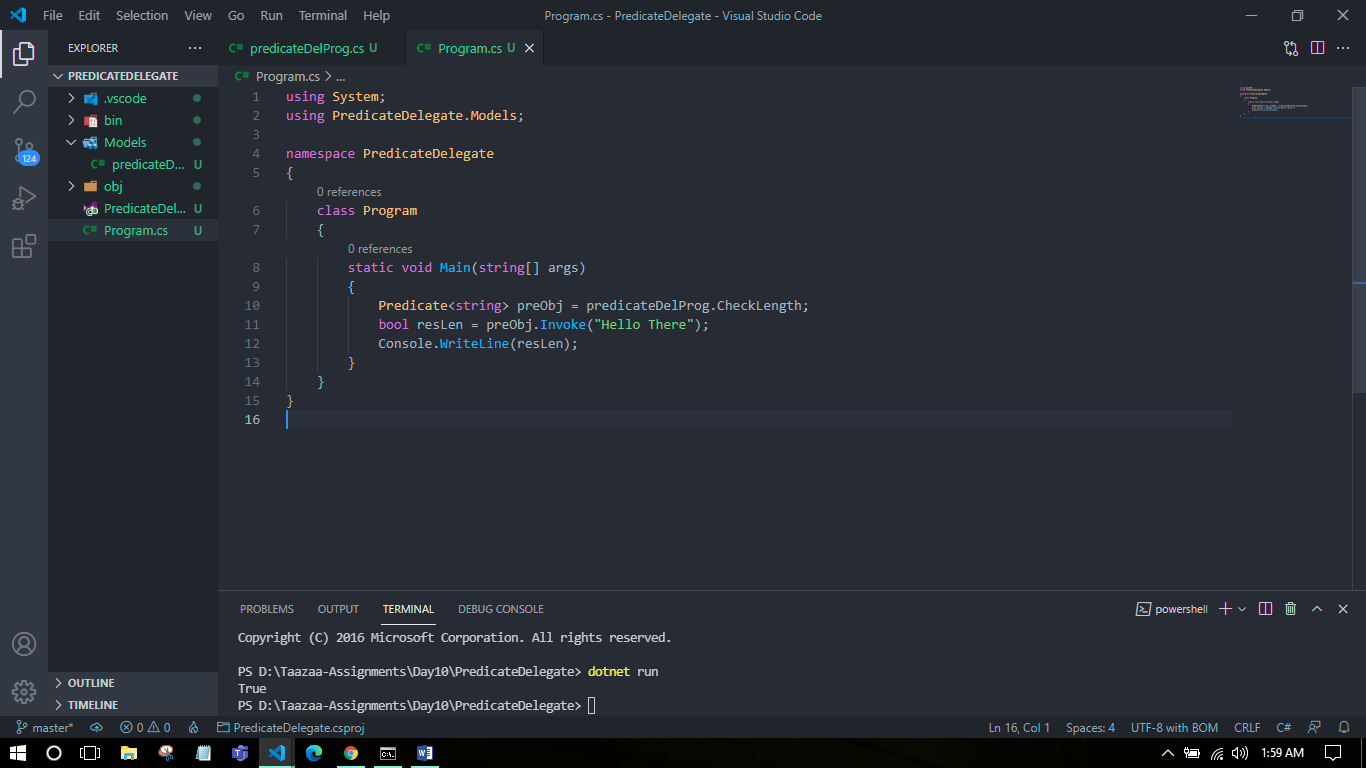
        }

    }

}

PS D:\Taazaa-Assignments\Day10\PredicateDelegate> dotnet run

True



5. Anonymous Method

using System;

public delegate string WelcomeDelegate(string name);

namespace AnonymousMethod

{

    class Program

    {

        static void Main()

        {

            WelcomeDelegate welcomeObj = delegate(string name)

            {

                return "Hello " + name + "! Welcome to the team.";

            };

            string res = welcomeObj.Invoke("Rahul");

            Console.WriteLine(res);

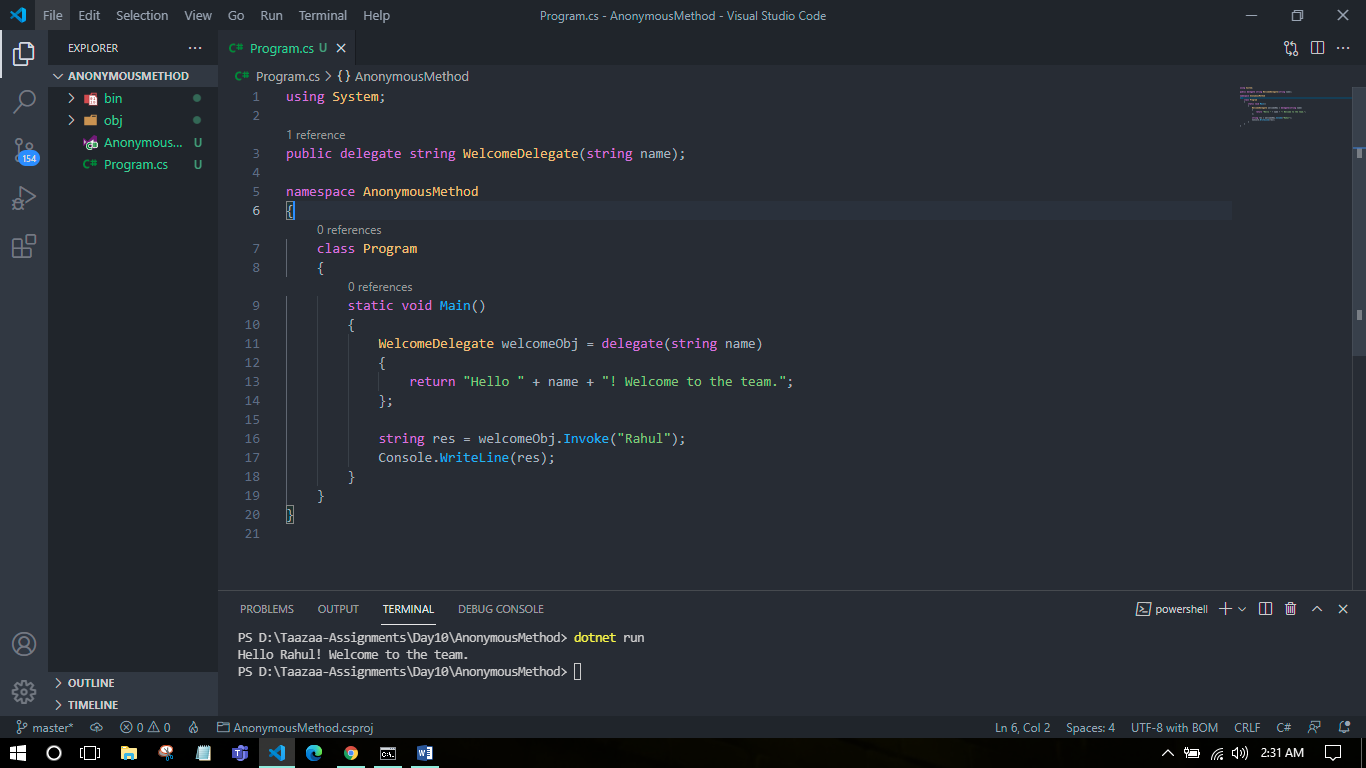
        }

    }

}

PS D:\Taazaa-Assignments\Day10\AnonymousMethod> dotnet run

Hello Rahul! Welcome to the team.



6. LINQ - LINQ provides the rich, standardized query syntax which allows the developers to interact with any data sources.

The responsibility of the LINQ provider is to convert the LINQ Query into a format so that the data source can understand it.

Different types of LINQ Objects: LINQ To Objects, LINQ To DataSets, LINQ To SQL, LINQ to XML, LINQ To Entities

namespace LinqQueries.Models

{

    public class LinqQuerProg

    {

        public int EmpId{get; set;}

        public string EmpName{get; set;}

        public string EmpProfile{get; set;}

    }

}

using System;

using System.Linq;

using System.Collections.Generic;

using LinqQueries.Models;

namespace LinqQueries

{

    class Program

    {

        static void Main()

        {

*// Student Collection*

            IList<LinqQuerProg> linqQuers = new List<LinqQuerProg>()

            {

                new LinqQuerProg() {EmpId = 101, EmpName = "Rahul", EmpProfile="ASET"},

                new LinqQuerProg() {EmpId = 102, EmpName = "Gurpreet", EmpProfile="AQAT"},

                new LinqQuerProg() {EmpId = 103, EmpName = "Sukhdev", EmpProfile="HR"},

                new LinqQuerProg() {EmpId = 104, EmpName = "Karan", EmpProfile="ASET"},

                new LinqQuerProg() {EmpId = 101, EmpName = "Divya", EmpProfile="HR"}

            };

            var EmpGroupByProfile = from p in linqQuers group p by p.EmpProfile into empPr orderby empPr.Key select new {empPr.Key, empPr};

            foreach(var group in EmpGroupByProfile){

                Console.WriteLine("{0} Profile:", group.Key);

                group.empPr.ToList().ForEach(em => Console.WriteLine(em.EmpName));

            }

        }

    }

}

PS D:\Taazaa-Assignments\Day10\LinqQueries> dotnet run

AQAT Profile:

Gurpreet

ASET Profile:

Rahul

Karan

HR Profile:

Sukhdev

Divya

