Definition:

should have a Point class which should have two public int variables x and y.
initialize it with 100 and 200
create another class Program
create a method static MyMethod which accept 2 parameters Point type p1 and p2
call from main method

p1.x +p2.x
p1.y +p2.y in method

Lab Ex

```
using System;


namespace DotNet
{
    class Point
    {
        public int x;
        public int y;
    }

    class Program
    {
        static void MyMethod(Point p1,Point p2){
                Console.WriteLine($"P1.X+p2.X = {p1.x+p2.x}");
                Console.WriteLine($"P1.Y+p2.Y ={p1.y+p2.y}");
        }
        static void Main(string[] args)
        {
            Point p1 = new Point();
            Point p2 = new Point();
            Console.Write("Enter Value Of P1 X :");
            p1.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P1 Y :");
            p1.y =  Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 X :");
            p2.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 Y :");
            p2.y =  Convert.ToInt32(Console.ReadLine());

            MyMethod(p1,p2);
        }
    }
}
```

Lab Ex Using ref Keyword

```
using System;


namespace DotNet
{
    class Point
    {
        public int x;
        public int y;
    }

    class Program
    {
        static void MyMethod(ref Point p1,ref Point p2){
                Console.WriteLine($"P1.X+p2.X = {p1.x+p2.x}");
                Console.WriteLine($"P1.Y+p2.Y ={p1.y+p2.y}");
        }
        static void Main(string[] args)
        {
            Point p1 = new Point();
            Point p2 = new Point();
            Console.Write("Enter Value Of P1 X :");
            p1.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P1 Y :");
            p1.y =  Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 X :");
            p2.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 Y :");
            p2.y =  Convert.ToInt32(Console.ReadLine());

            MyMethod(ref p1,ref p2);
        }
    }
}
```

Lab Ex Using out Keyword

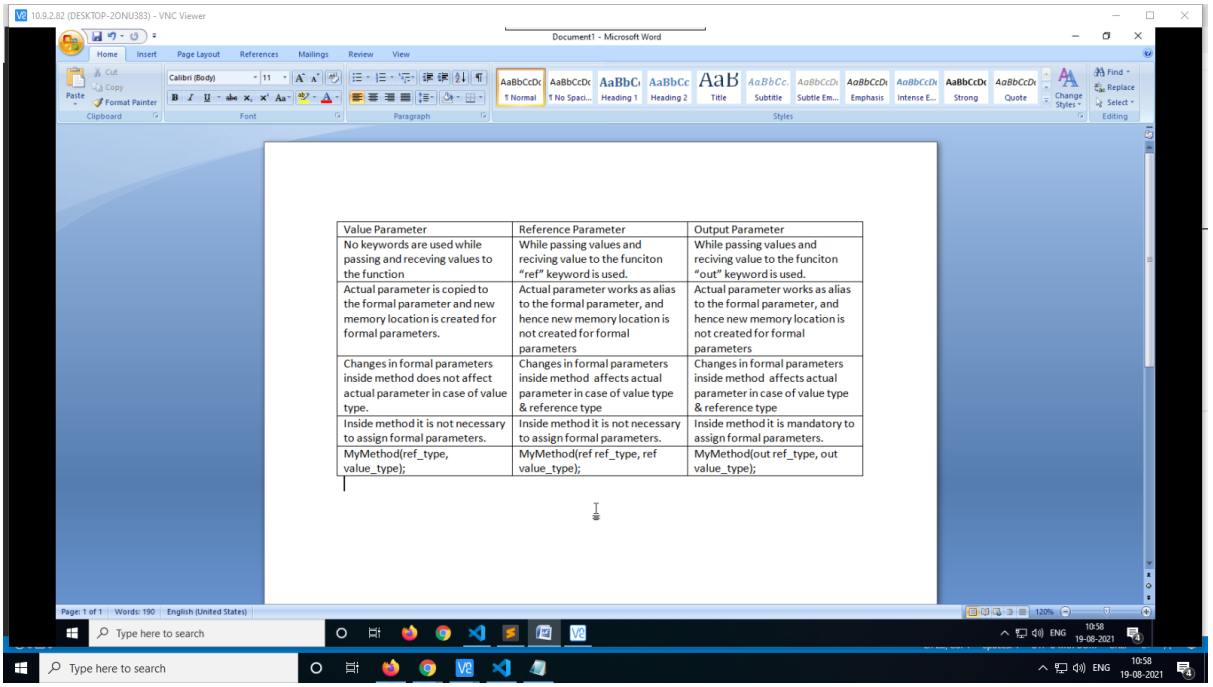
```
using System;


namespace DotNet
{
    class Point
    {
        public int x;
        public int y;
    }

    class Program
    {
        static void MyMethod(out Point p1,out Point p2){
            p1 = new Point();
            p2 = new Point();
            Console.Write("Enter Value Of P1 X :");
            p1.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P1 Y :");
            p1.y =  Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 X :");
            p2.x = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter Value Of P2 Y :");
            p2.y =  Convert.ToInt32(Console.ReadLine());

            Console.WriteLine($"P1.X+p2.X = {p1.x+p2.x}");
            Console.WriteLine($"P1.Y+p2.Y ={p1.y+p2.y}");
        }
        static void Main(string[] args)
        {
            Point p1 =null;
            Point p2 = null;

            MyMethod(out p1,out p2);
        }
    }
}
```

Parameter Array

```
using System;


namespace DotNet
{


    class Program
    {
        static void Sum(params int[] values){
            int total=0;
            for(int i=0;i<values.Length;i++){
                total +=values[i];
            }
            Console.WriteLine($"{total}");
        }
        static void PrintName(params string[] names){
            for(int i=0;i<names.Length;i++){
                Console.WriteLine($"{names[i]}");
            }
        }

        static void Main(string[] args)
        {
            Sum(10,20,30);
            PrintName("Ankit","Rudani");
        }
    }
}
```

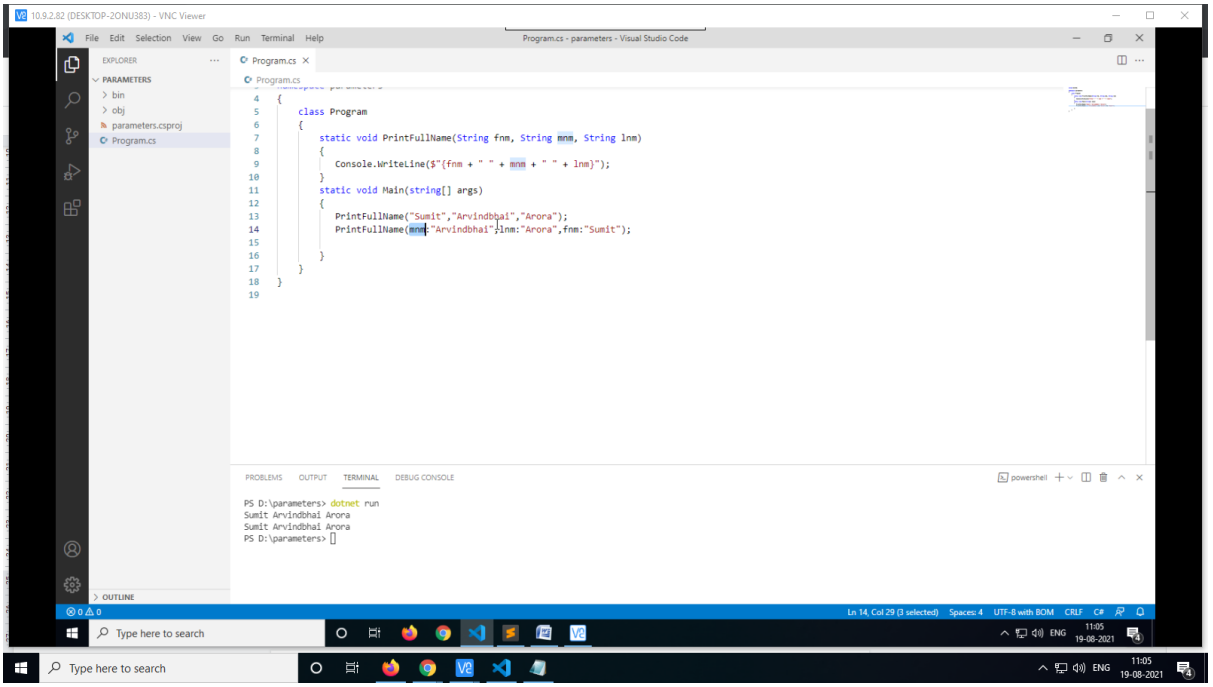
Parameter Array find Square and Cube

```
using System;

namespace DotNet
{
    class Program
    {
        static void SquareCube(params int[] nos){
            Console.WriteLine("----------------------------------");
            for(int i=0;i<nos.Length;i++){
                Console.WriteLine($"{nos[i]}^2 = {nos[i]*nos[i]} ,
{nos[i]}^3 = {nos[i]*nos[i]*nos[i]}");
            }
        }

        static void Main(string[] args)
        {
            SquareCube(1,2,3,4);
            SquareCube(5,6,7,8);
        }
    }
}
```

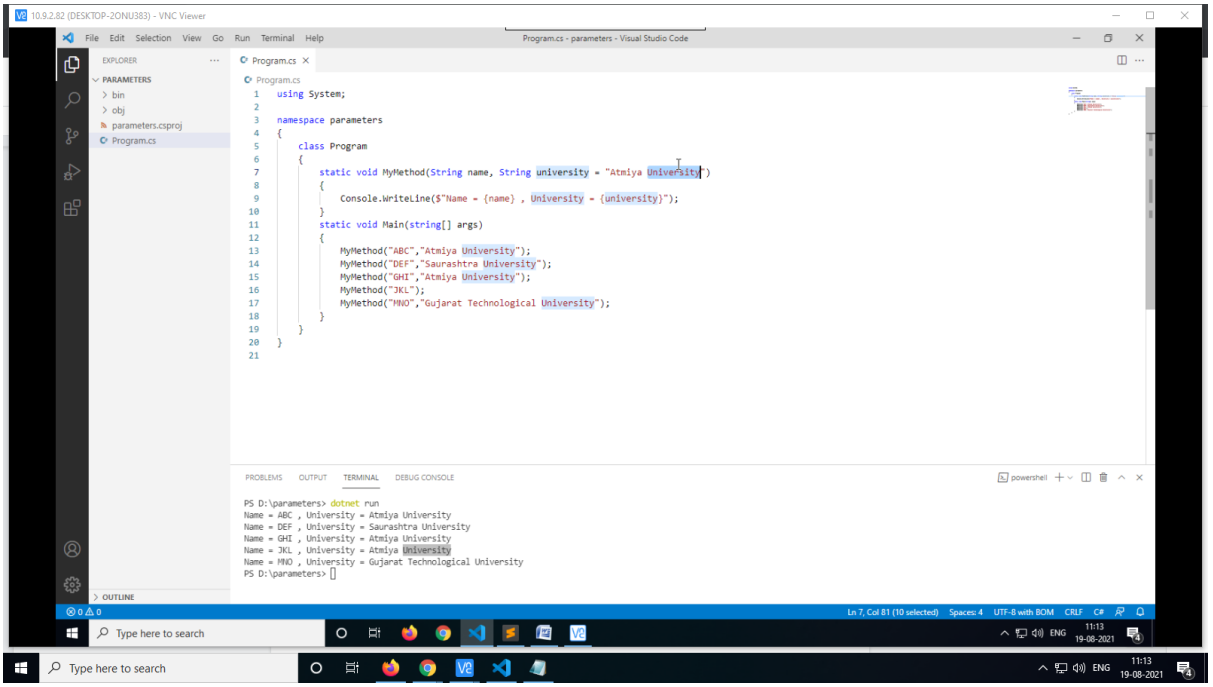


Value Parameter	Reference Parameter	Output Parameter
No keywords are used while passing and receving values to the function	While passing values and reciving value to the funciton “ref” keyword is used.	While passing values and reciving value to the funciton “out” keyword is used.
Actual parameter is copied to the formal parameter and new memory location is created for formal parameters.	Actual parameter works as alias to the formal parameter, and hence new memory location is not created for formal parameters	Actual parameter works as alias to the formal parameter, and hence new memory location is not created for formal parameters
Changes in formal parameters inside method does not affect actual parameter in case of value type.	Changes in formal parameters inside method affects actual parameter in case of value type & reference type	Changes in formal parameters inside method affects actual parameter in case of value type & reference type
Inside method it is not necessary to assign formal parameters.	Inside method it is not necessary to assign formal parameters.	Inside method it is mandatory to assign formal parameters.
MyMethod(ref_type, value_type);	MyMethod(ref ref_type, ref value_type);	MyMethod(out ref_type, out value_type);

Named Parameters



Optional Parameters



Definition :

Write a C# Program to Create a class Employee which has two public variables name, designation and salary (int)
Program Method DispalyDetails(Employee e1)
if Designation is not pass then by default assign "Jr.SW"

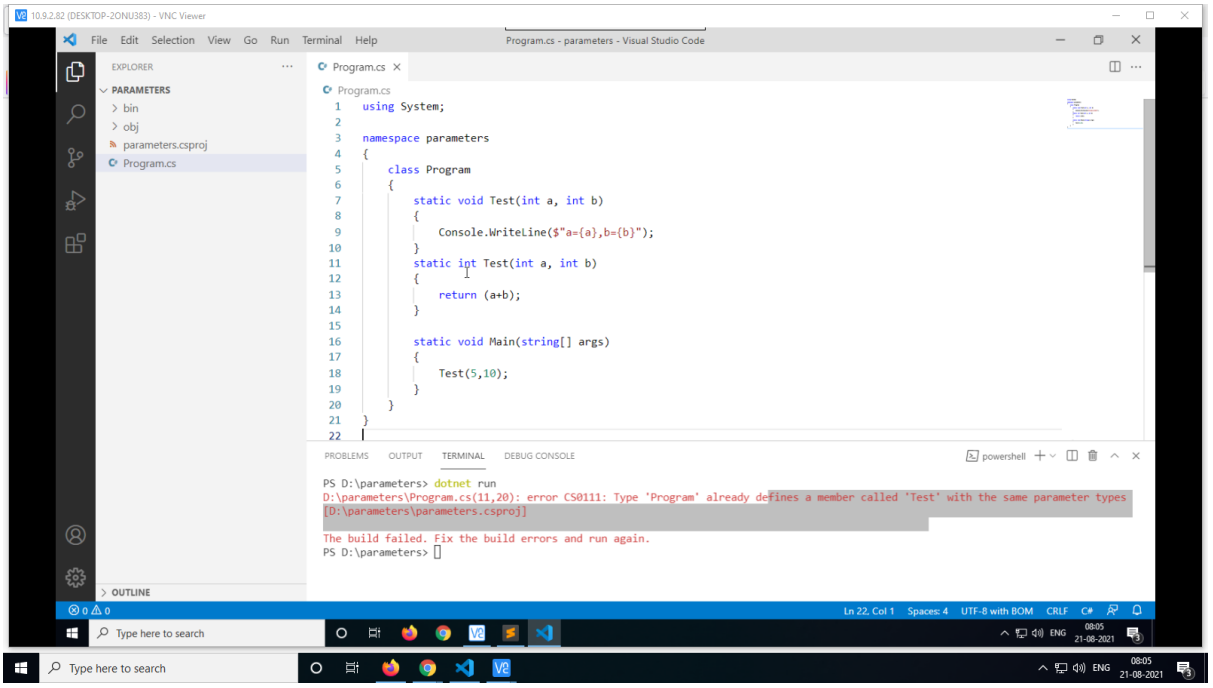
```
using System;


namespace DotNet
{
    class Employee
    {
        public string name,designation="Jr. Software Developer";
        public int salary;
    }
    class Program
    {
        static void DisplayDetail(Employee e)
        {
            Console.WriteLine($"Name = {e.name}, Designation =
{e.designation}, Salary = {e.salary}");
        }
        static void Main(string[] args)
        {
            Employee e1 = new Employee();
            e1.name = "Ankit";
            e1.designation = "DBA";
            e1.salary = 1000;
            DisplayDetail(e1);

            Employee e2 = new Employee();
            e2.name = "Ankur";
            e2.salary = 1000;
            DisplayDetail(e2);
        }
    }
}
```

Method Overloading:

When a class have more than one method with same name is called method overloading, but they should have different signatures.

Method signatures includes:

1. Name of the Method
2.

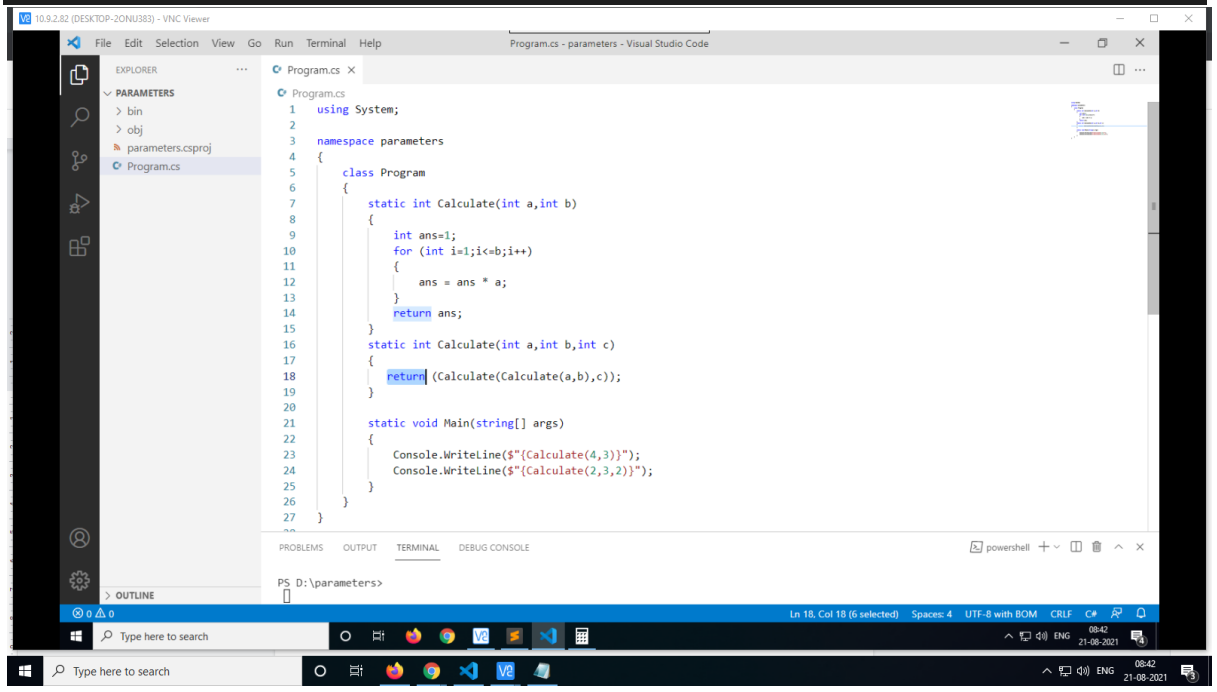


Definition:

Write a C# program to overload a calculate method which is overloaded with no of parameters is to parameters are pass it will return a^b
if three parameters are passed it will return (a^b)^c

```
using System;

namespace DotNet
{
    class Program
    {
        static int Calculate(int a,int b){
            int ans = 1;
            for(int i=0;i<b;i++)
                ans *=a;
            return ans;
        }
```

```
        static int Calculate(int a,int b,int c){// Overloaded Method
            int ans = Calculate(a,b);
            return Calculate(ans,c);
        }
        static void Main(string[] args)
        {
            Console.WriteLine($"{Calculate(2,3)}");
            Console.WriteLine($"{Calculate(2,3,4)}");
        }
    }
}
```



Static Constructor

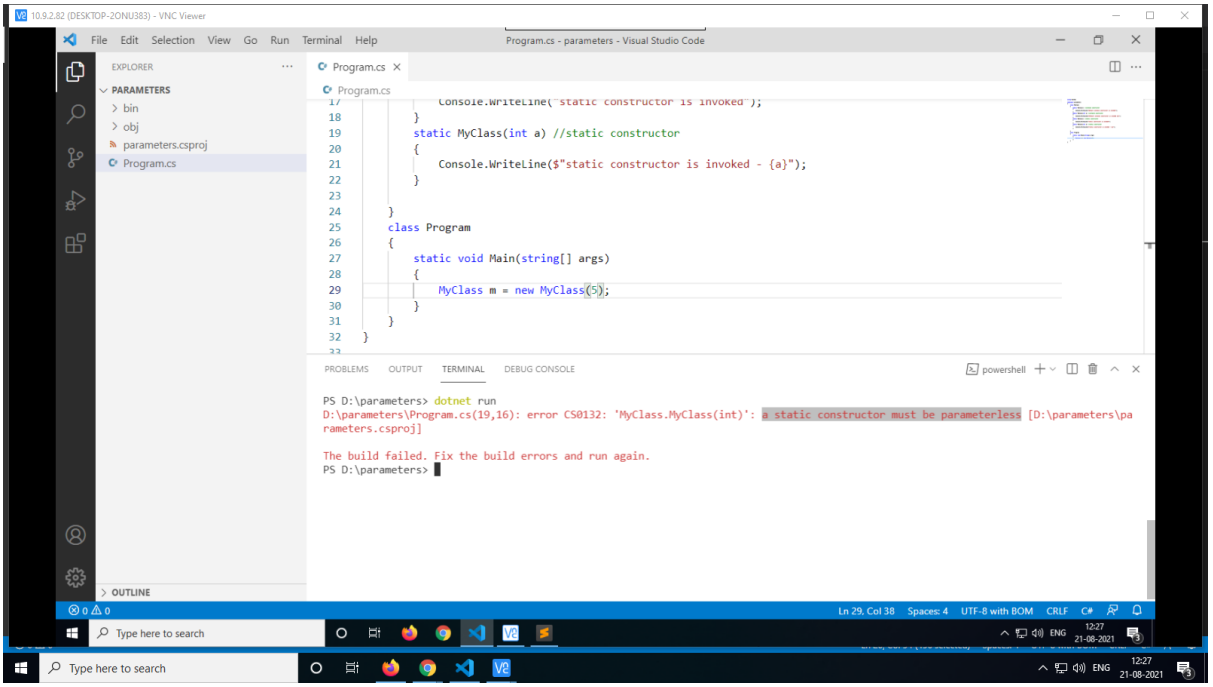
Definition :

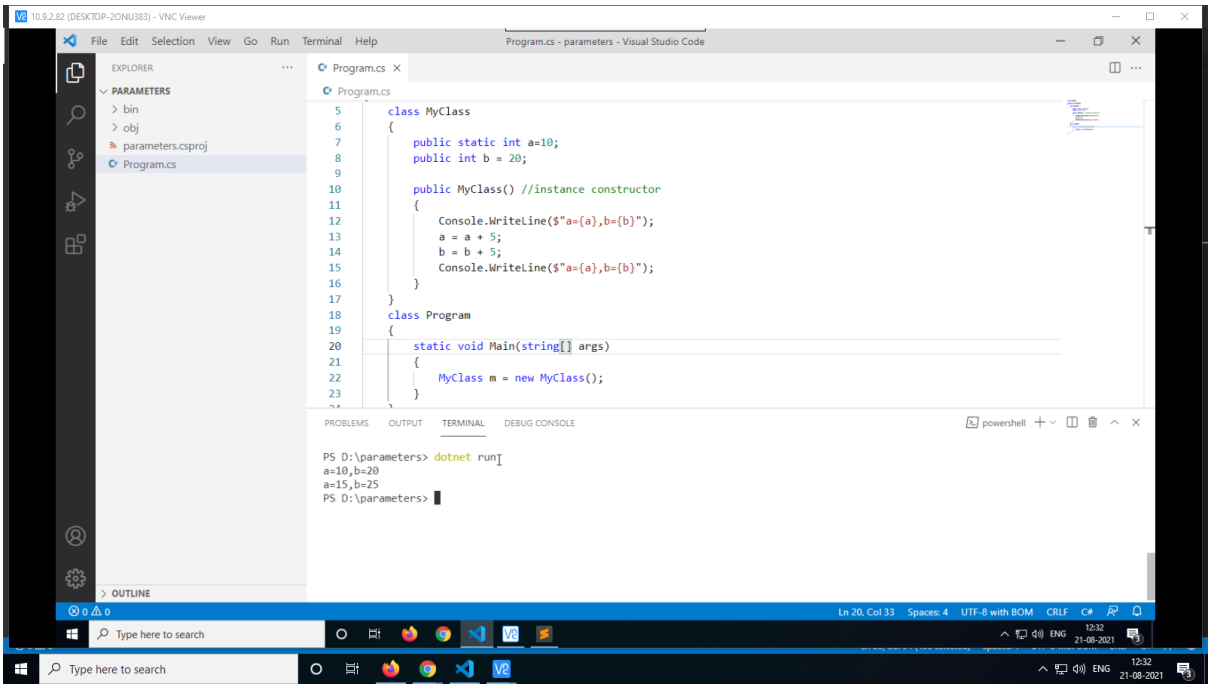
create a class Employee which is the static variable noe and non static members name and salary
create a parameterized constructor which takes name and salary as parameter to initialize instance members.
when new employee instance is created noe variable should be ++
class should have method to get name and salary and to print the name and salary and
a method to print no of employees.

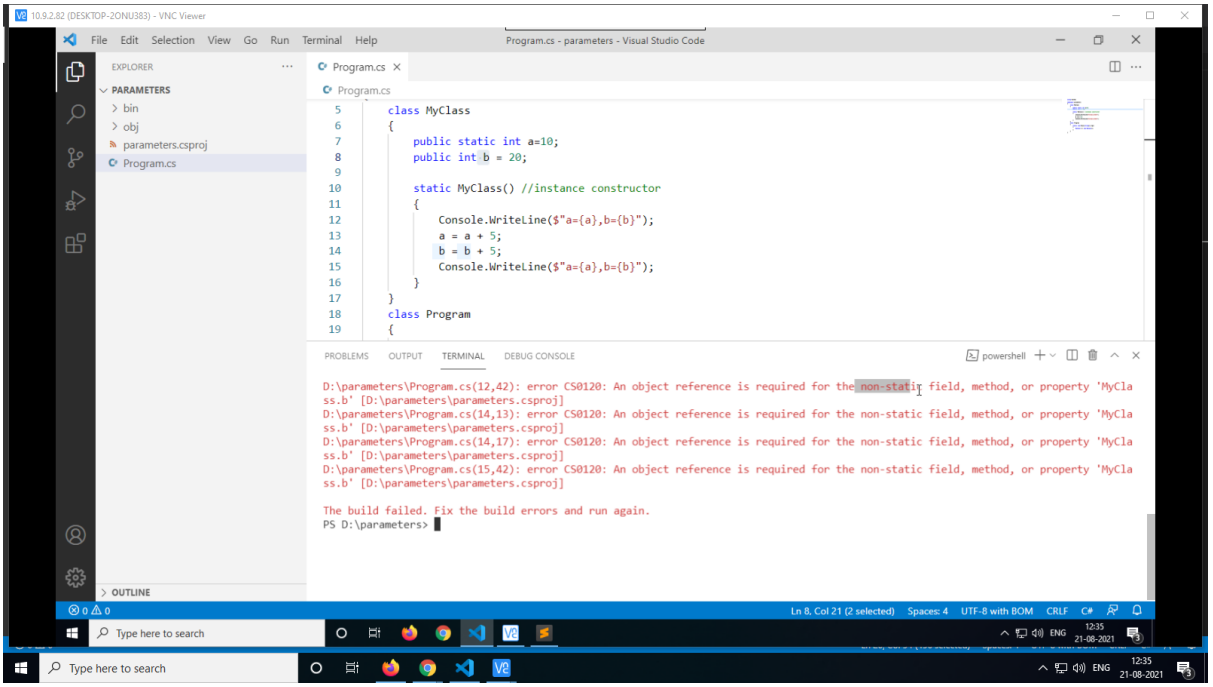
Answer

```
using System;

namespace DotNet
{
    class Employee{
        public static int noe=0;
        public string name;
        public int salary;

        public Employee(){
            name = "abc";
            salary = 12000;
            noe++;
        }

        public Employee(string nm,int sal){
            name = nm;
            salary = sal;
            noe++;
        }

        // static Employee(){
        //     noe++;
        //     Console.WriteLine("Hello");
        // }

        public void getDetails(){
            Console.WriteLine($"Name = {name}\nSalary = {salary}");
            Console.WriteLine("------------------------------");
```

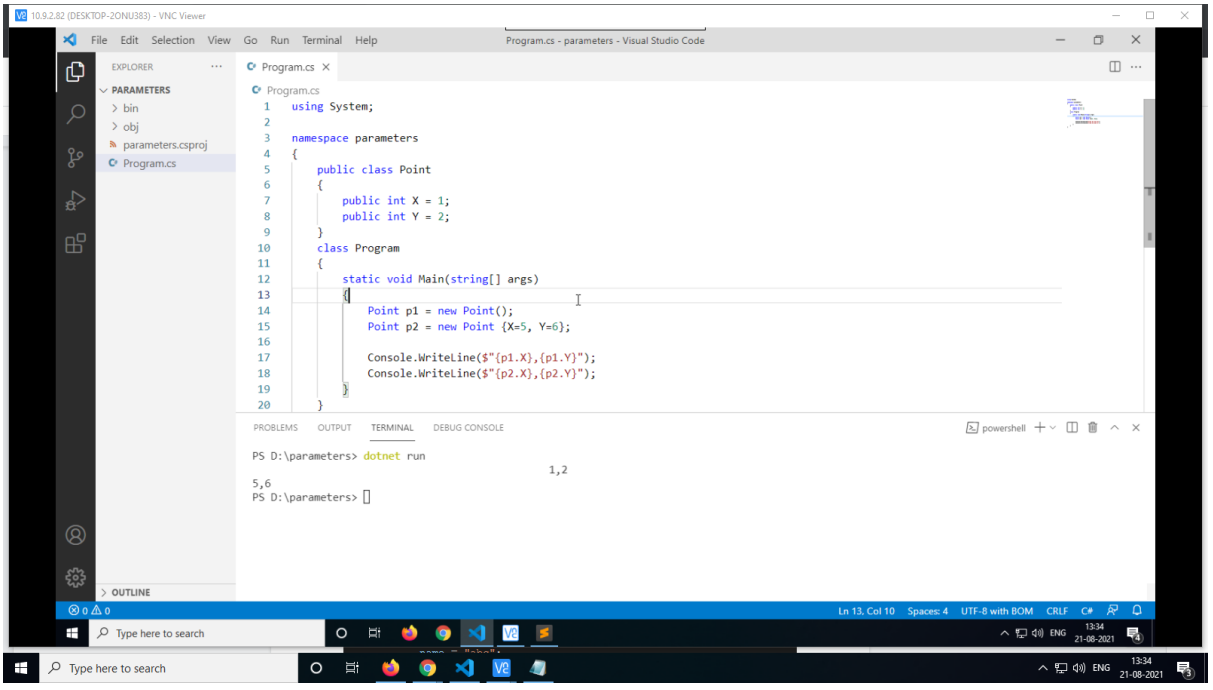
```
        }

        public static void getNOE(){
            Console.WriteLine($"No of Employee = {noe}");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Employee e1 = new Employee();
            e1.getDetails();

            Employee e2 = new Employee("Ankit",10000);
            e2.getDetails();
            Employee.getNOE();

            Employee e3 = new Employee("Ankur",15000);
            e3.getDetails();
            //Employee.getNOE();

            Employee e4 = new Employee("Kaushal",20000);
            e4.getDetails();
            Employee.getNOE();
        }
    }
}
```


Program.cs - parameters - Visual Studio Code
Program.cs
static MyClass(int a) //static constructor
{
Console.WriteLine($"static constructor is invoked - {a}");
}
class Program
{
static void Main(string[] args)
{
MyClass m = new MyClass(5);
}
}
PS D:\parameters> dotnet run
D:\parameters\Program.cs(19,16): error CS0132: 'MyClass.MyClass(int)': a static constructor must be parameterless [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>

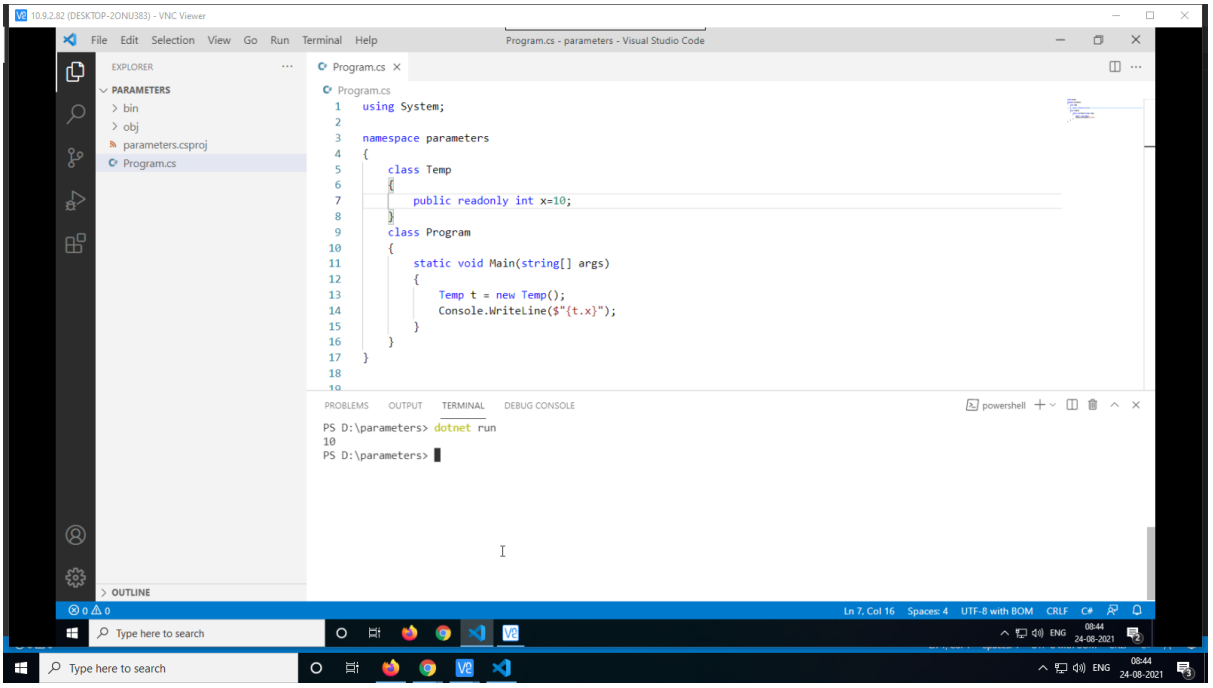

Program.cs - parameters - Visual Studio Code
Program.cs
class MyClass
{
public static int a=10;
public int b = 20;
public MyClass() //instance constructor
{
Console.WriteLine($"a={a},b={b}");
a = a + 5;
b = b + 5;
Console.WriteLine($"a={a},b={b}");
}
}
class Program
{
static void Main(string[] args)
{
MyClass m = new MyClass();
}
PS D:\parameters> dotnet run
a=10,b=20
a=15,b=25
PS D:\parameters>

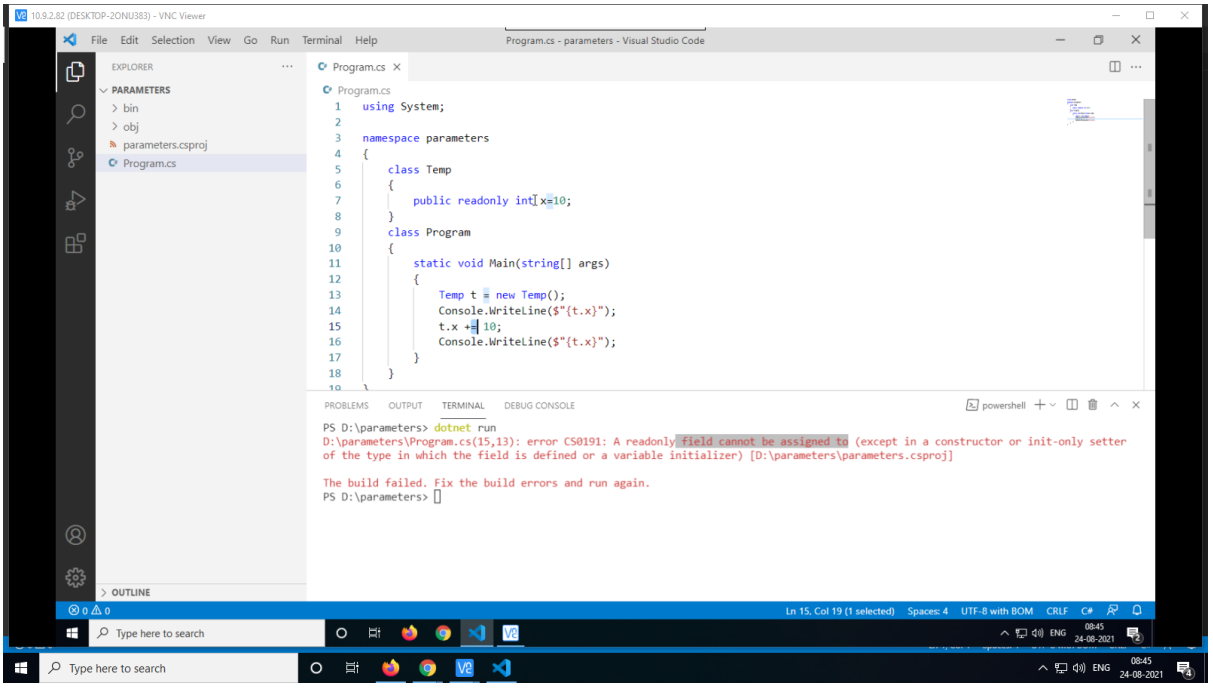


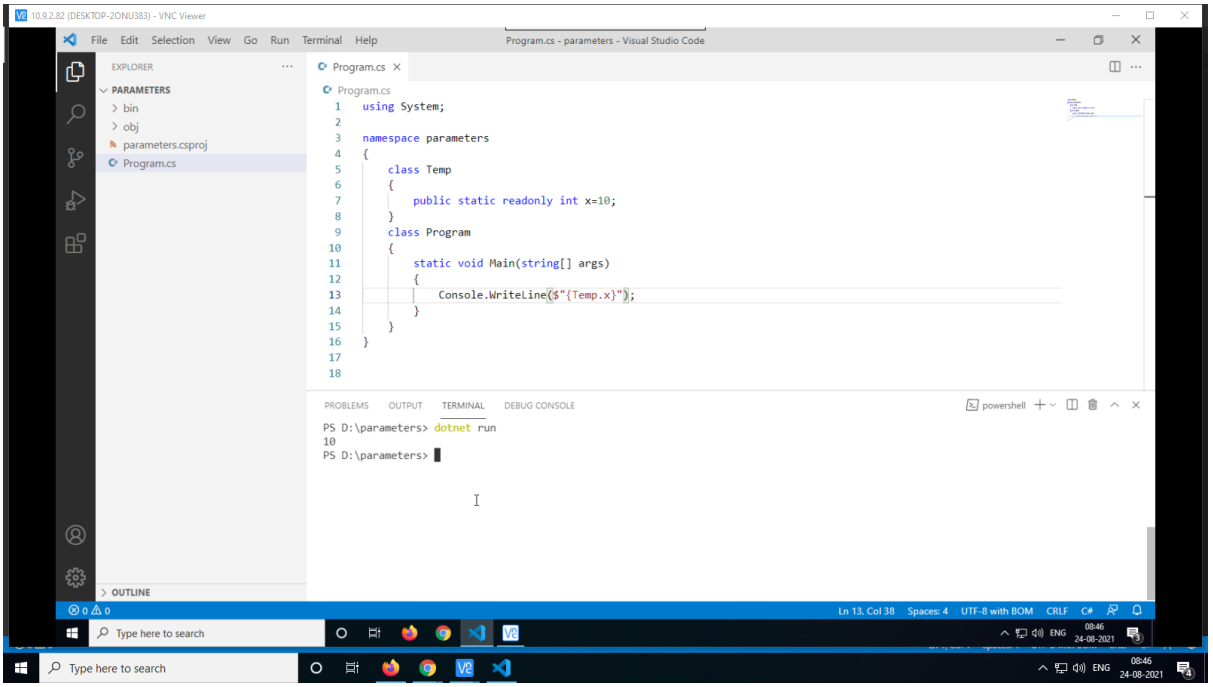
Object Initializer

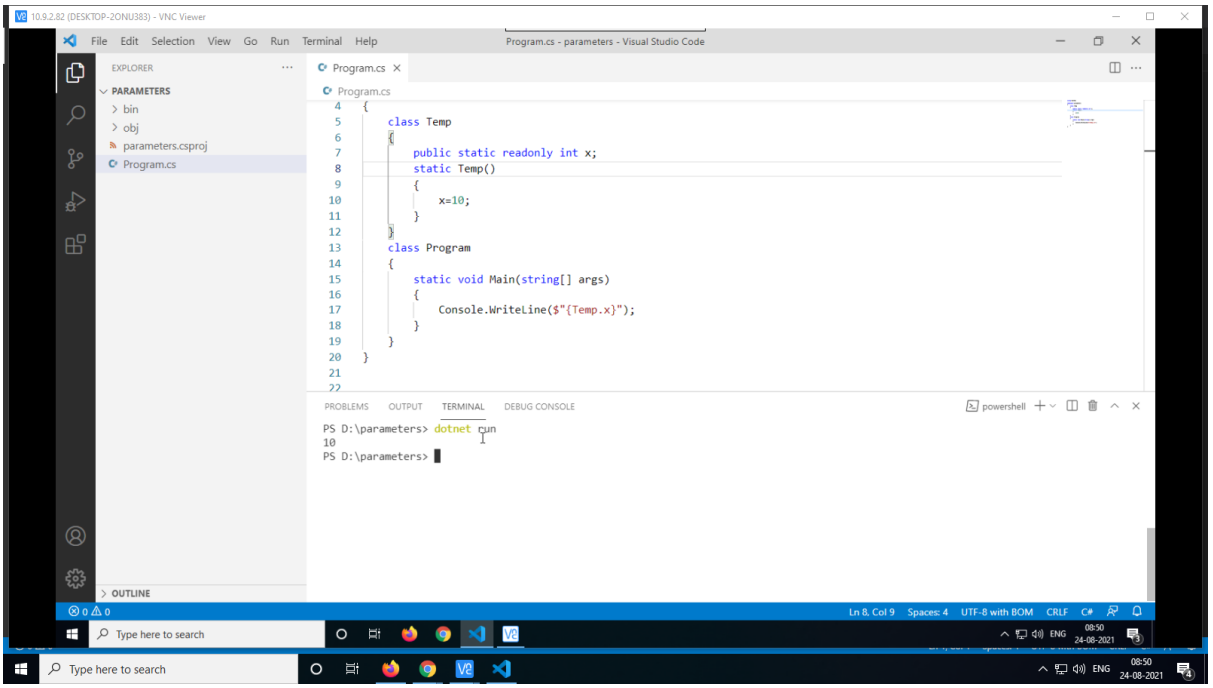


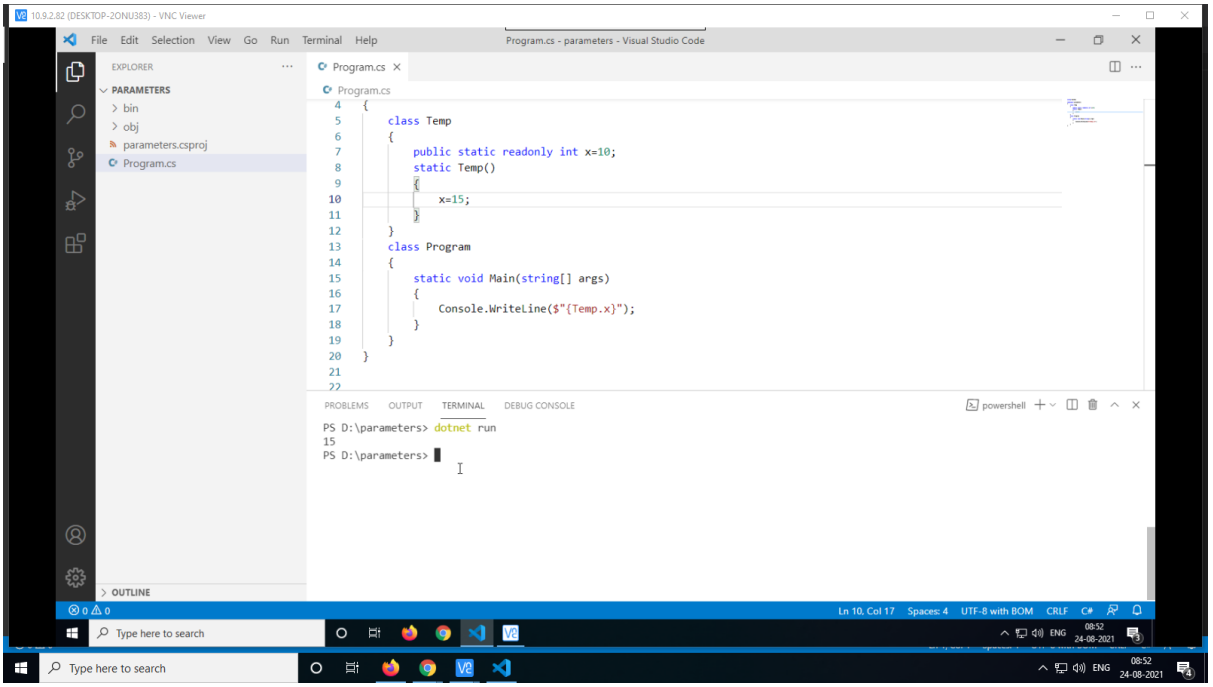
Readonly

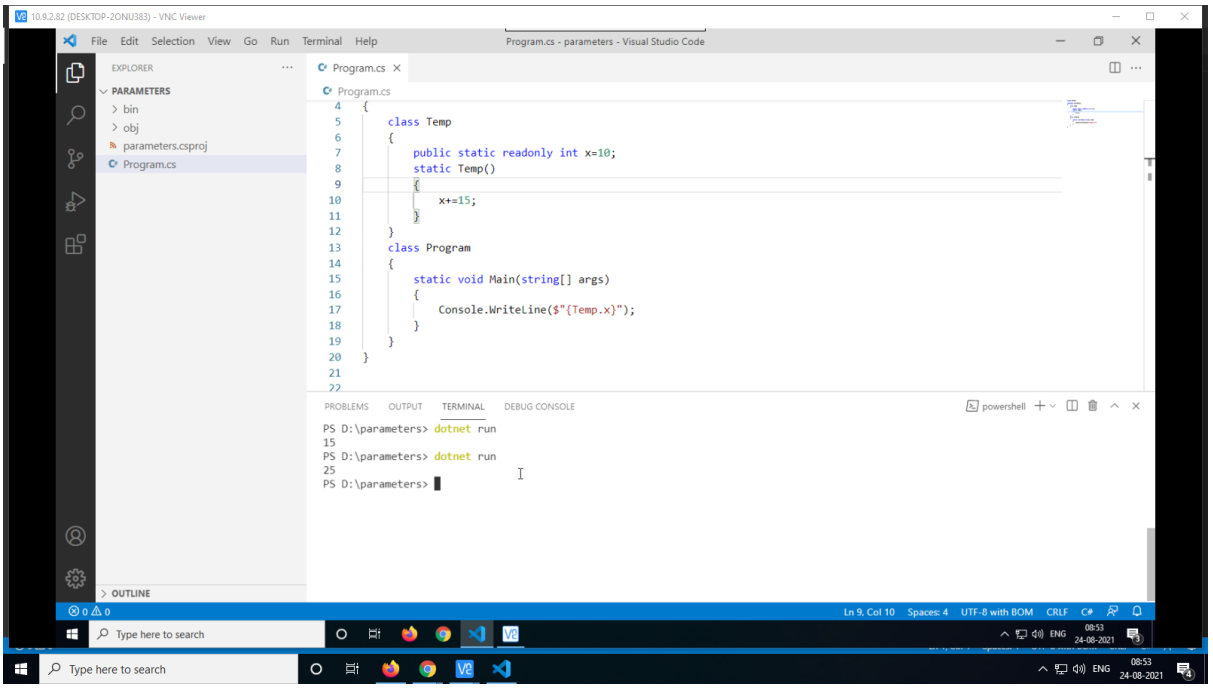

Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
using System;
namespace parameters
{
class Temp
{
public readonly int x=10;
}
class Program
{
static void Main(string[] args)
{
Temp t = new Temp();
Console.WriteLine($"{t.x}");
}
}
}
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
PS D:\parameters> dotnet run
10
PS D:\parameters>
OUTLINE
Ln 7, Col 16
Spaces: 4
UTF-8 with BOM
CRLF
C#
Type here to search

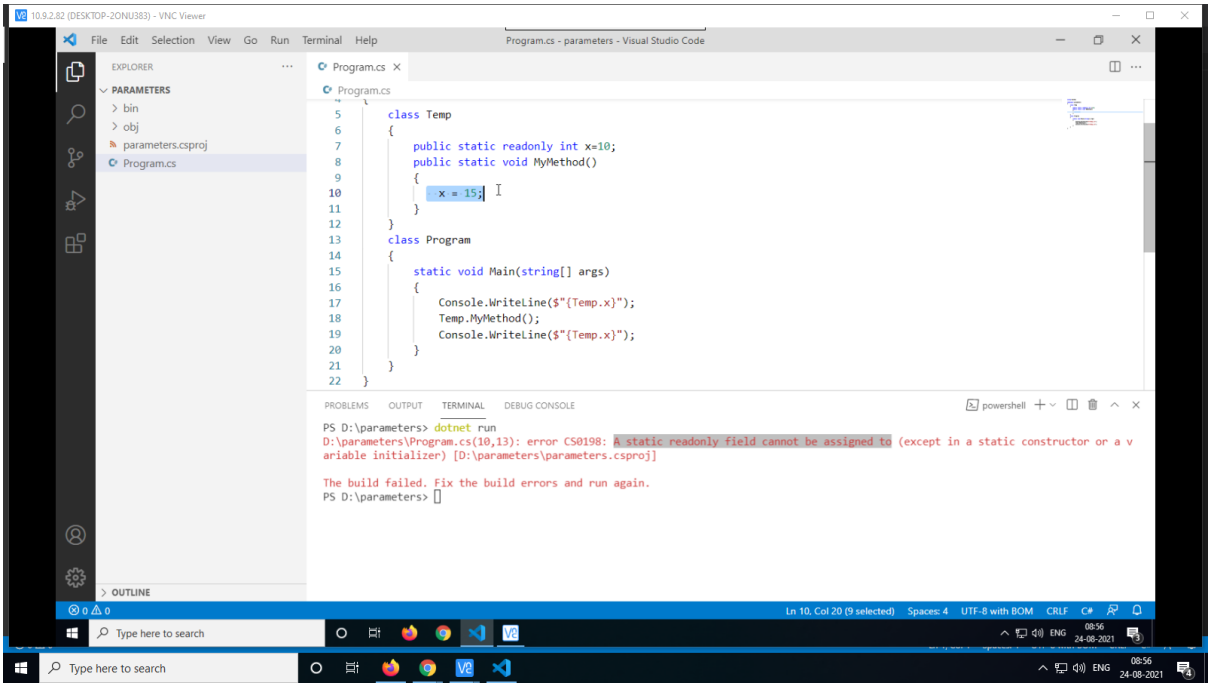

Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
using System;
namespace parameters
{
class Temp
{
public readonly int x=10;
}
class Program
{
static void Main(string[] args)
{
Temp t = new Temp();
Console.WriteLine($"{t.x}");
t.x += 10;
Console.WriteLine($"{t.x}");
}
}
}
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
PS D:\parameters> dotnet run
D:\parameters\Program.cs(15,13): error CS0191: A readonly field cannot be assigned to (except in a constructor or init-only setter of the type in which the field is defined or a variable initializer) [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>
OUTLINE
Ln 15, Col 19 (1 selected)
Spaces: 4
UTF-8 with BOM
CRLF
C#
Type here to search

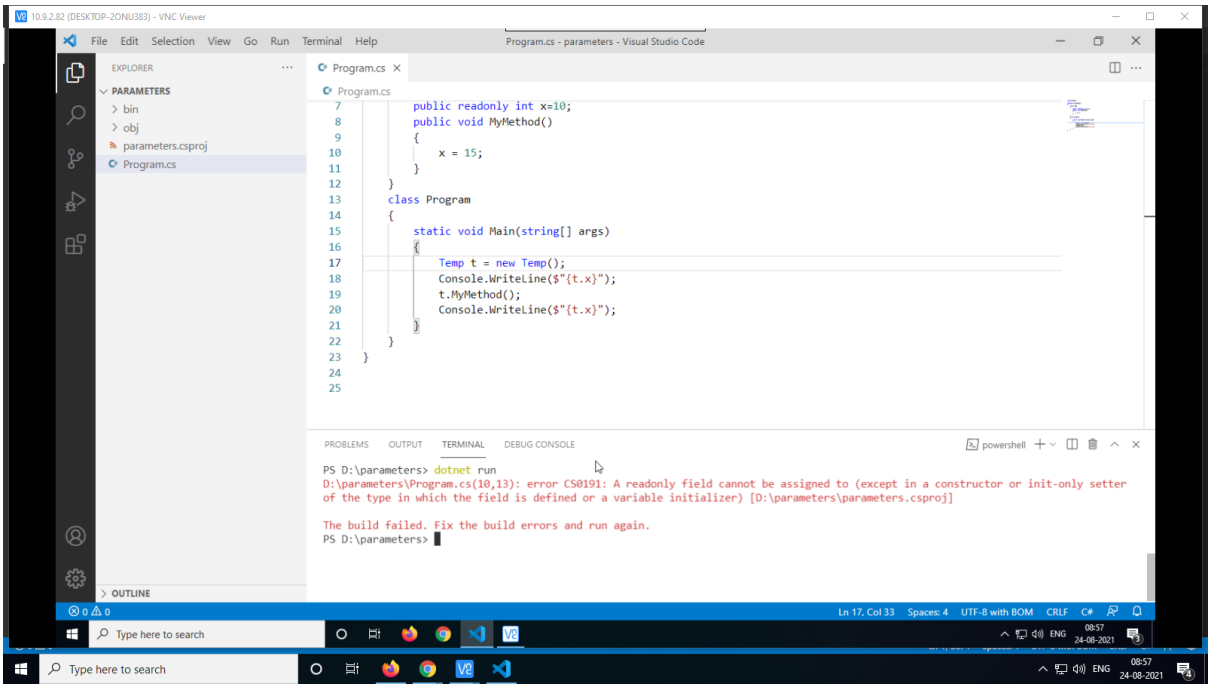

10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
using System;
namespace parameters
{
class Temp
{
public static readonly int x=10;
}
class Program
{
static void Main(string[] args)
{
Console.WriteLine($"{Temp.x}");
}
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
10
PS D:\parameters>
OUTLINE
Ln 13, Col 38 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG
08:46
24-08-2021

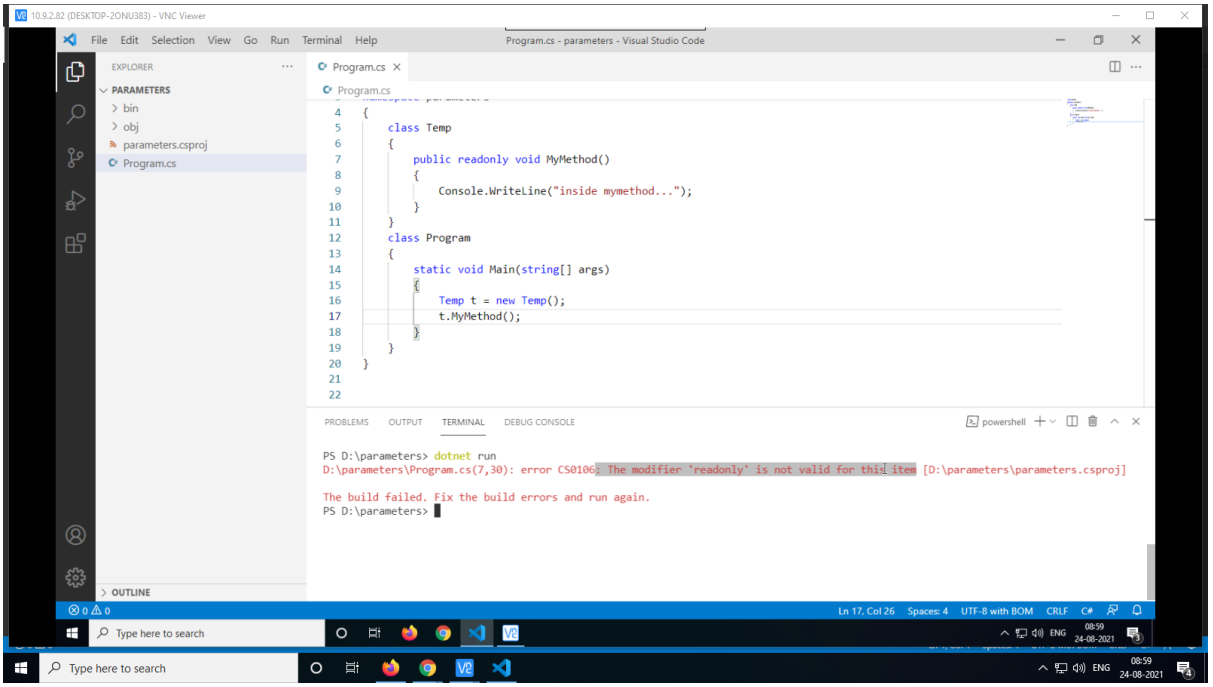

10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
{
class Temp
{
public static readonly int x;
static Temp()
{
x=10;
}
}
class Program
{
static void Main(string[] args)
{
Console.WriteLine($"{Temp.x}");
}
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
10
PS D:\parameters>
OUTLINE
Ln 8, Col 9 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG
08:50
24-08-2021

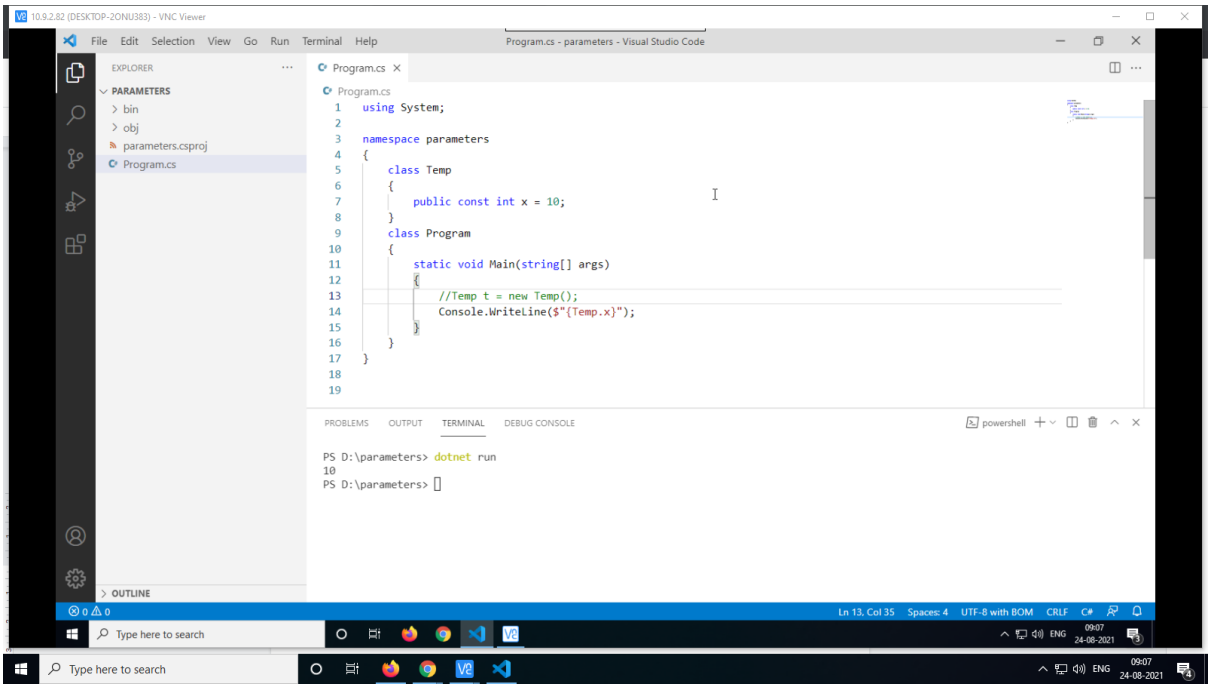

10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
4 {
5 class Temp
6 {
7 public static readonly int x=10;
8 static Temp()
9 {
10 x=15;
11 }
12 }
13 class Program
14 {
15 static void Main(string[] args)
16 {
17 Console.WriteLine($"{Temp.x}");
18 }
19 }
20 }
21
22
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
15
PS D:\parameters>
OUTLINE
Ln 10, Col 17 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 08:52 24-08-2021

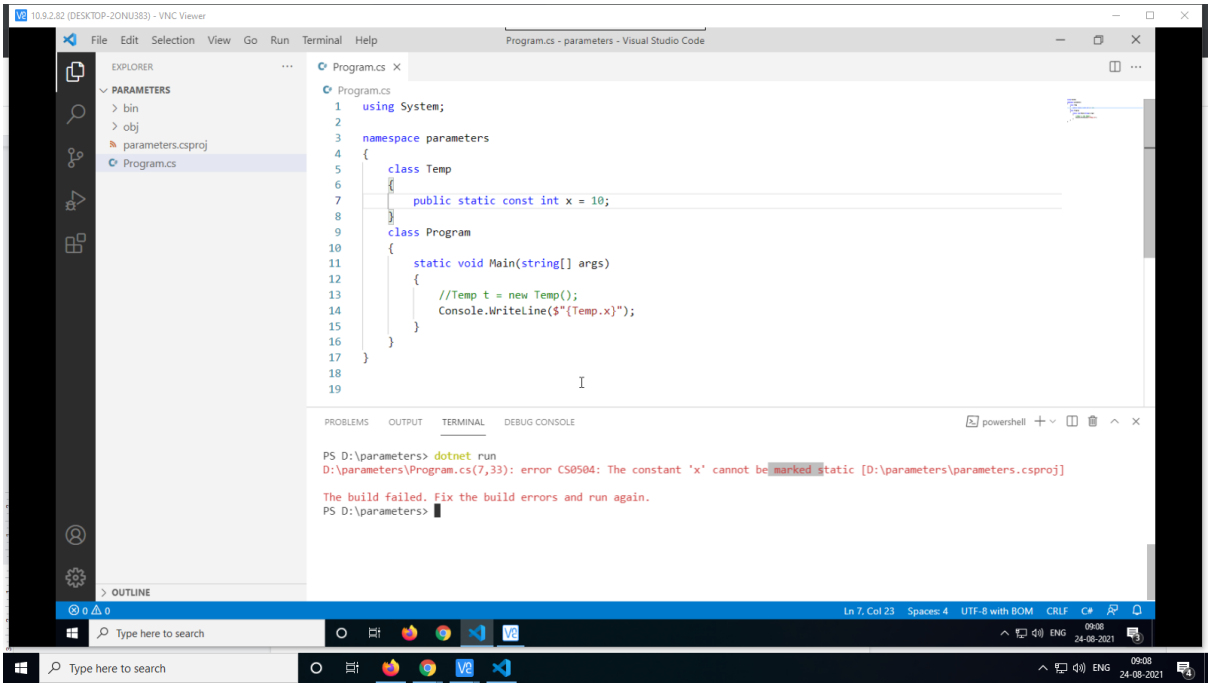

10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
4 {
5 class Temp
6 {
7 public static readonly int x=10;
8 static Temp()
9 {
10 x+=15;
11 }
12 }
13 class Program
14 {
15 static void Main(string[] args)
16 {
17 Console.WriteLine($"{Temp.x}");
18 }
19 }
20 }
21
22
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
15
PS D:\parameters> dotnet run
25
PS D:\parameters>
OUTLINE
Ln 9, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 08:53 24-08-2021

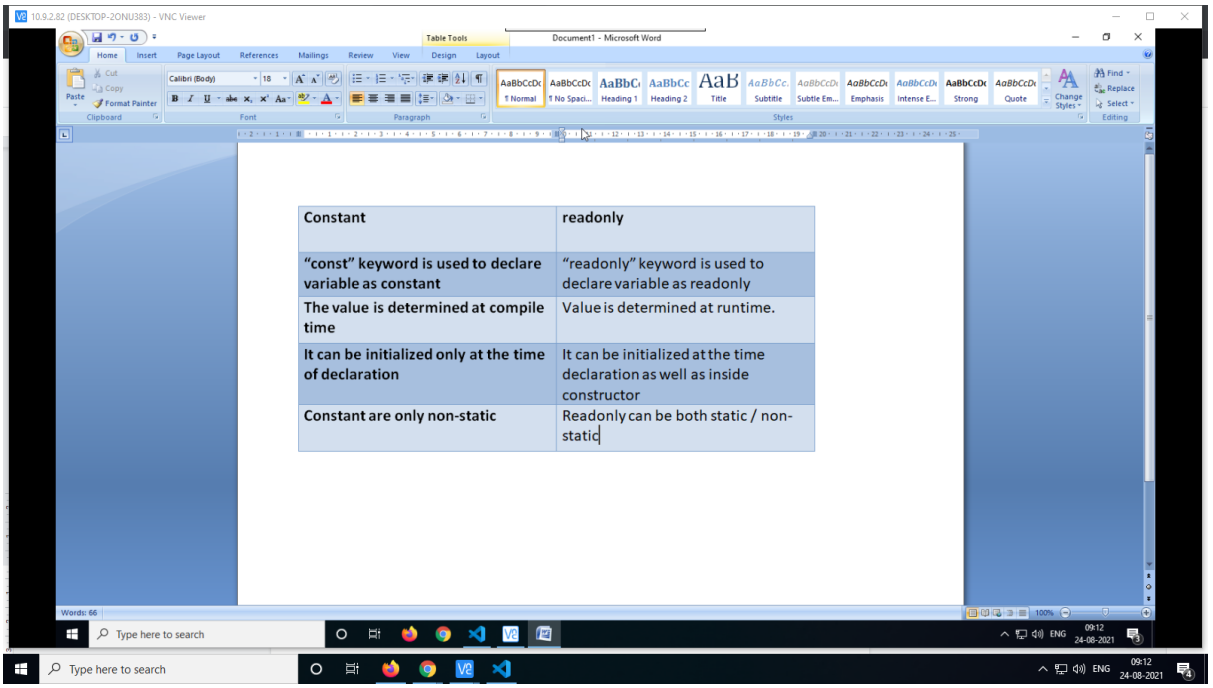

Program.cs - parameters - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
class Temp
{
public static readonly int x=10;
public static void MyMethod()
{
x = 15;
}
}
class Program
{
static void Main(string[] args)
{
Console.WriteLine($"{Temp.x}");
Temp.MyMethod();
Console.WriteLine($"{Temp.x}");
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
D:\parameters\Program.cs(10,13): error CS0198: A static readonly field cannot be assigned to (except in a static constructor or a variable initializer) [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>
OUTLINE
Ln 10, Col 20 (9 selected) Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 08:56 24-08-2021

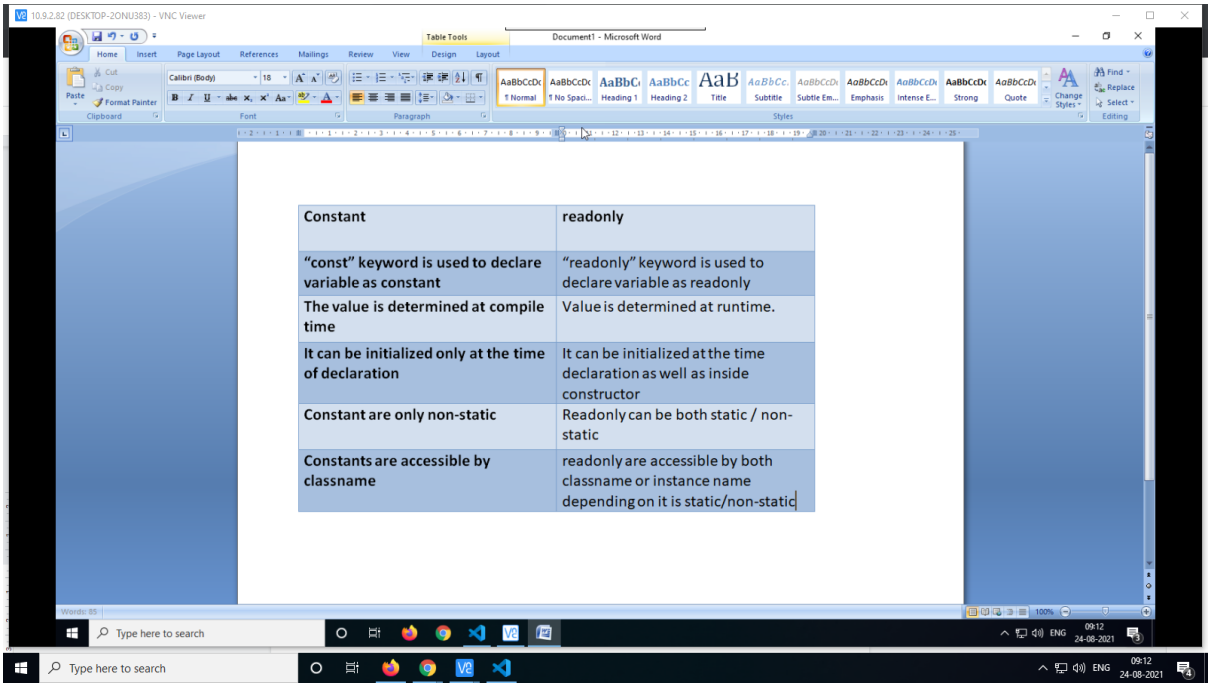

Program.cs - parameters - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
public readonly int x=10;
public void MyMethod()
{
x = 15;
}
}
class Program
{
static void Main(string[] args)
{
Temp t = new Temp();
Console.WriteLine($"{t.x}");
t.MyMethod();
Console.WriteLine($"{t.x}");
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
D:\parameters\Program.cs(10,13): error CS0191: A readonly field cannot be assigned to (except in a constructor or init-only setter of the type in which the field is defined or a variable initializer) [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>
OUTLINE
Ln 17, Col 33 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 08:57 24-08-2021


10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
4 {
5 class Temp
6 {
7 public readonly void MyMethod()
8 {
9 Console.WriteLine("inside mymethod...");
10 }
11 }
12 class Program
13 {
14 static void Main(string[] args)
15 {
16 Temp t = new Temp();
17 t.MyMethod();
18 }
19 }
20 }
21
22
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
D:\parameters\Program.cs(7,30): error CS0106: The modifier 'readonly' is not valid for this item [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>
OUTLINE
Ln 17, Col 26 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 08:59 24-08-2021

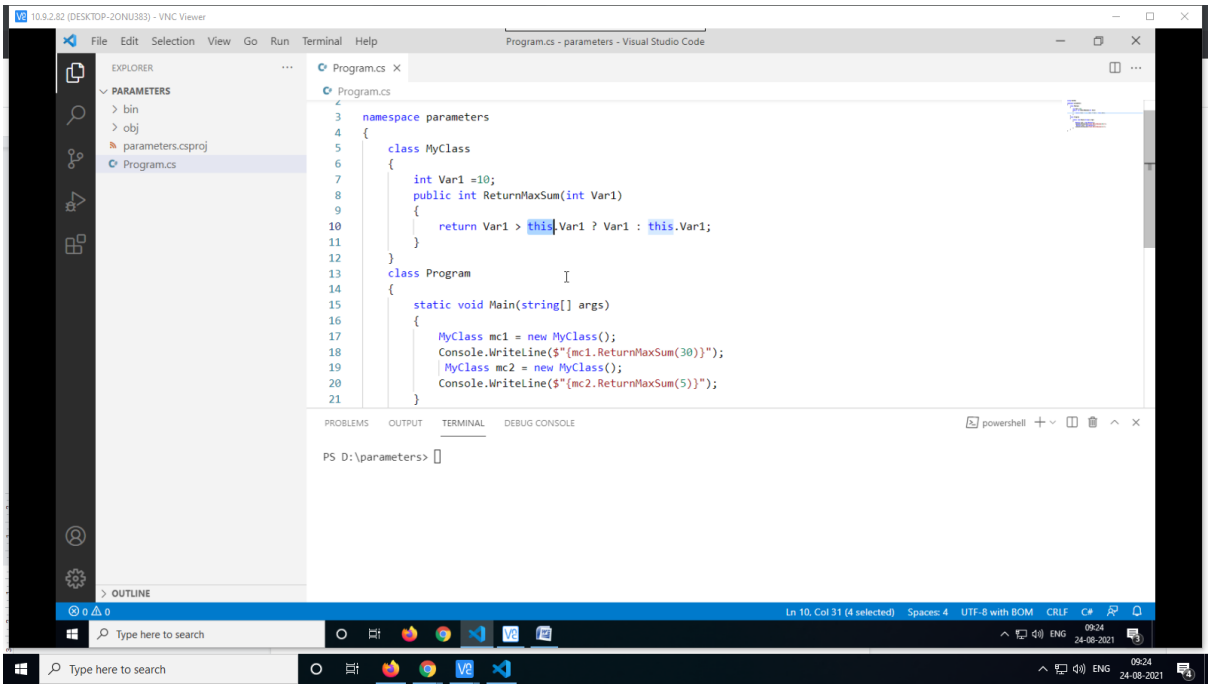

10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
1 using System;
2
3 namespace parameters
4 {
5 class Temp
6 {
7 public const int x = 10;
8 }
9 class Program
10 {
11 static void Main(string[] args)
12 {
13 //Temp t = new Temp();
14 Console.WriteLine($"{Temp.x}");
15 }
16 }
17 }
18
19
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
10
PS D:\parameters>
OUTLINE
Ln 13, Col 35 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 09:07 24-08-2021


Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
using System;
namespace parameters
{
class Temp
{
public static const int x = 10;
}
class Program
{
static void Main(string[] args)
{
//Temp t = new Temp();
Console.WriteLine($"{Temp.x}");
}
}
}
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
D:\parameters\Program.cs(7,33): error CS0504: The constant 'x' cannot be marked static [D:\parameters\parameters.csproj]
The build failed. Fix the build errors and run again.
PS D:\parameters>
OUTLINE
Ln 7, Col 23
Spaces: 4
UTF-8 with BOM
CRLF
C#
Type here to search
ENG
09:08
24-08-2021


Document1 - Microsoft Word
Table Tools
Constant
readonly
"const" keyword is used to declare variable as constant
"readonly" keyword is used to declare variable as readonly
The value is determined at compile time
Value is determined at runtime.
It can be initialized only at the time of declaration
It can be initialized at the time declaration as well as inside constructor
Constant are only non-static
Readonly can be both static / non-static
Words: 66
Type here to search
ENG
09:12
24-08-2021



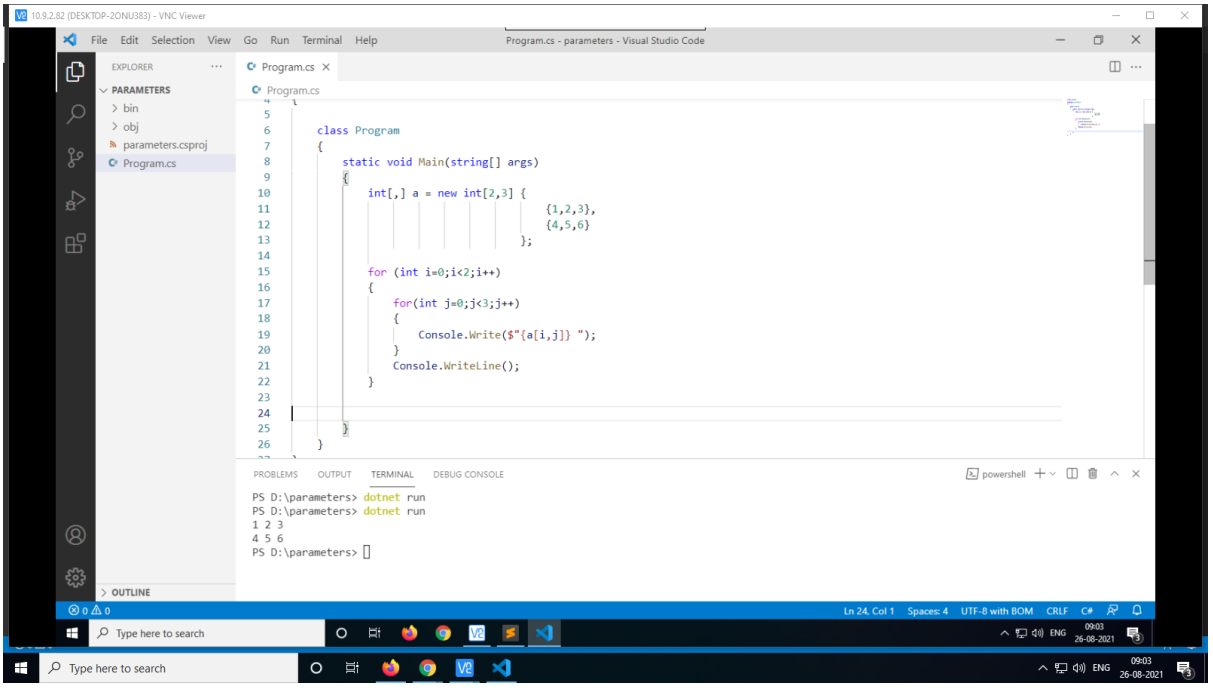
Constant	readonly
"const" keyword is used to declare variable as constant	"readonly" keyword is used to declare variable as readonly
The value is determined at compile time	Value is determined at runtime.
It can be initialized only at the time of declaration	It can be initialized at the time declaration as well as inside constructor
Constant are only non-static	Readonly can be both static / non-static
Constants are accessible by classname	readonly are accessible by both classname or instance name depending on it is static/non-static

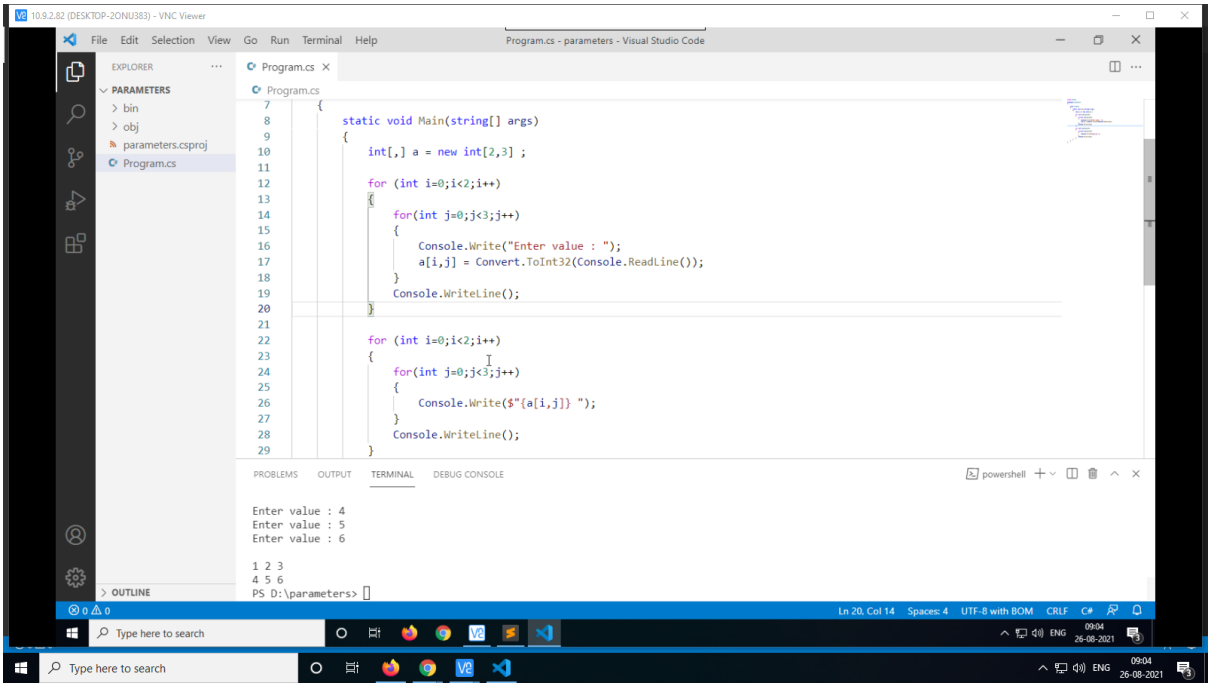


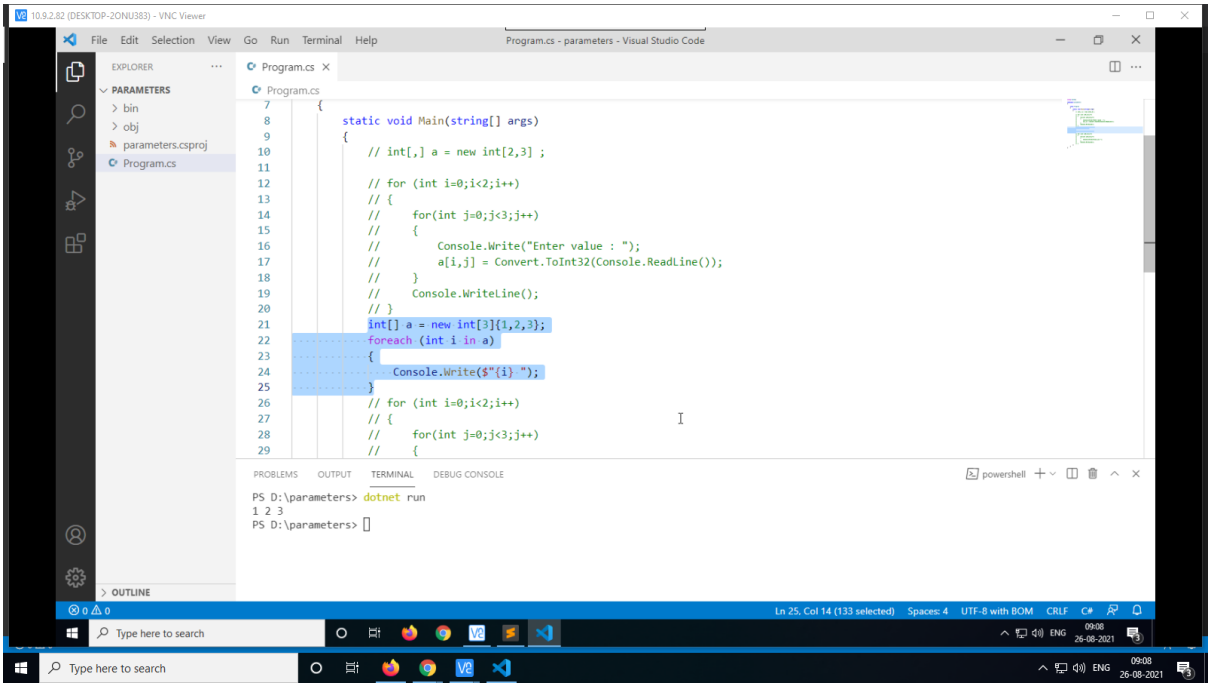
```
namespace parameters
{
    class MyClass
    {
        int Var1 =10;
        public int ReturnMaxSum(int Var1)
        {
            return Var1 > this.Var1 ? Var1 : this.Var1;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            MyClass mc1 = new MyClass();
            Console.WriteLine($"{mc1.ReturnMaxSum(30)}");
            MyClass mc2 = new MyClass();
            Console.WriteLine($"{mc2.ReturnMaxSum(5)}");
        }
```

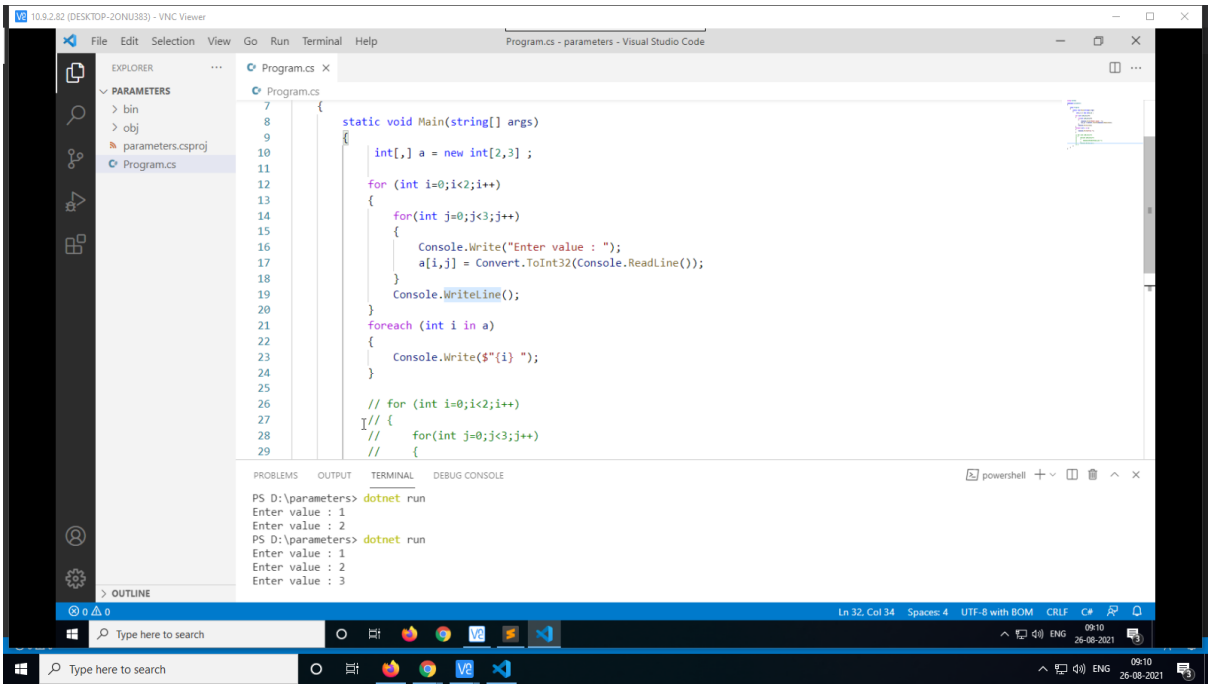
```
PS D:\parameters>
```

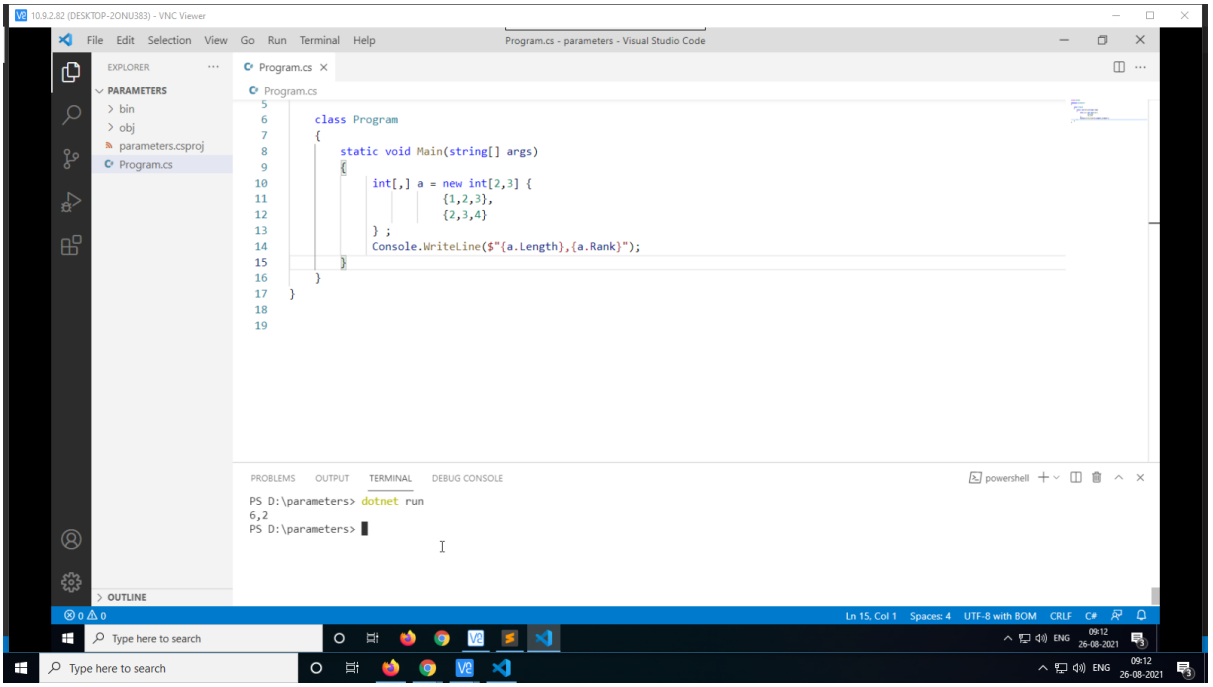
Rectangular Array

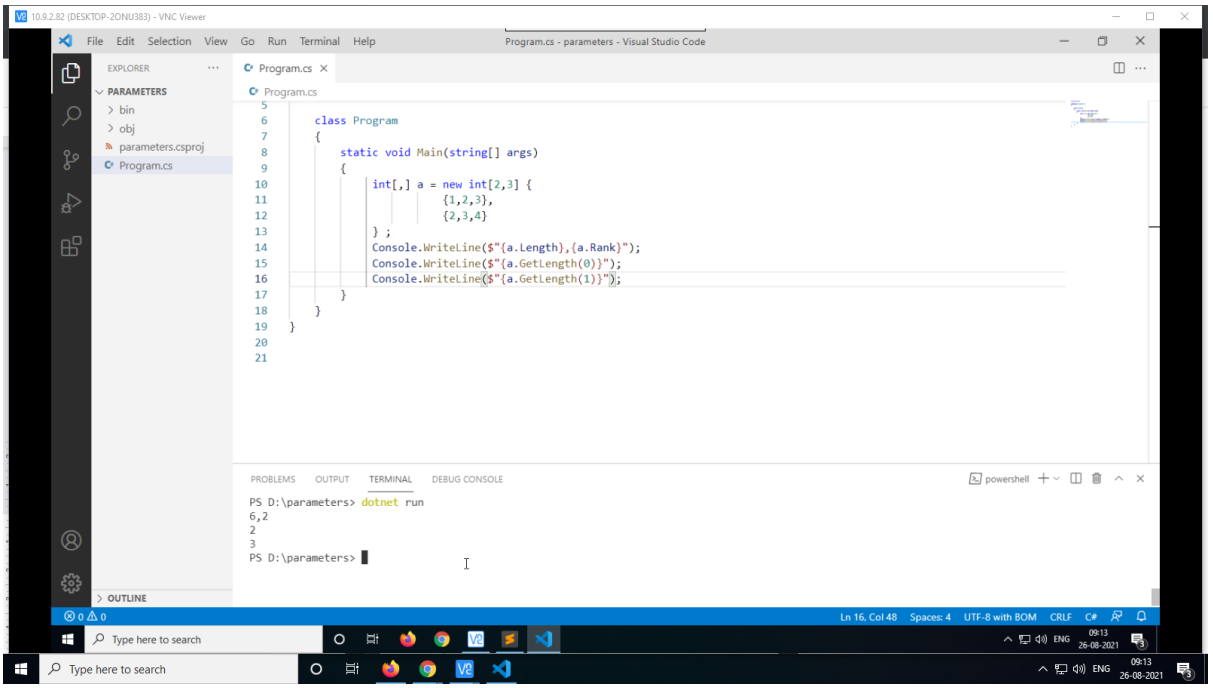


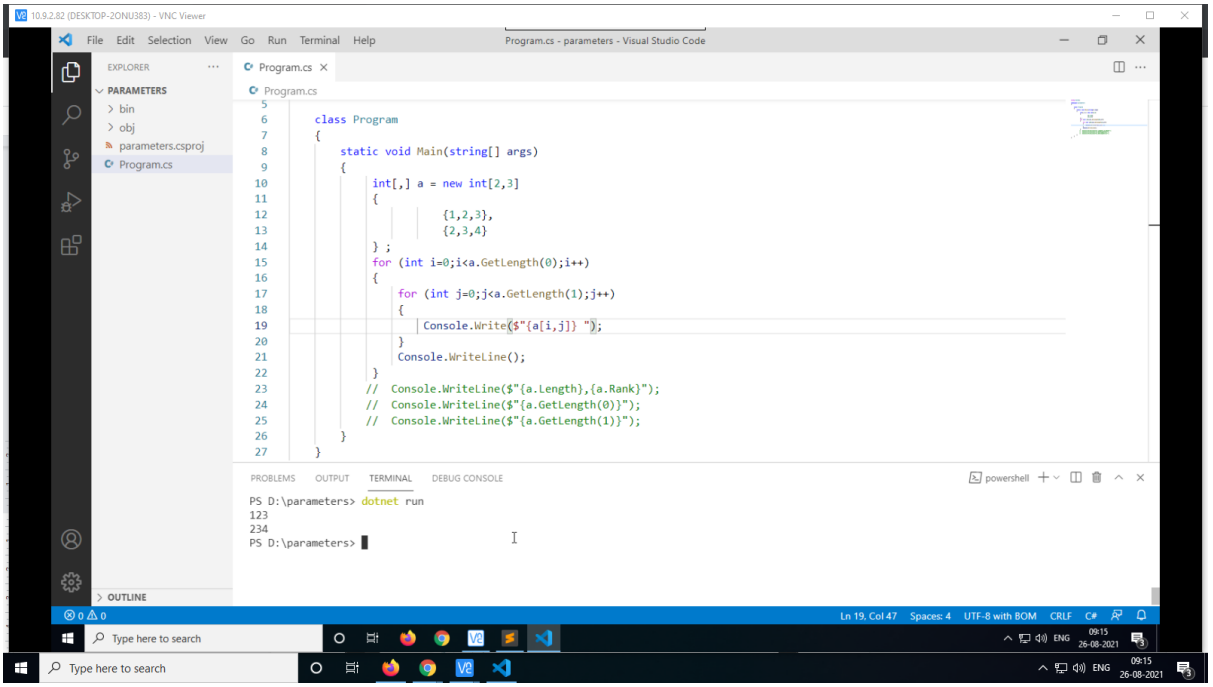


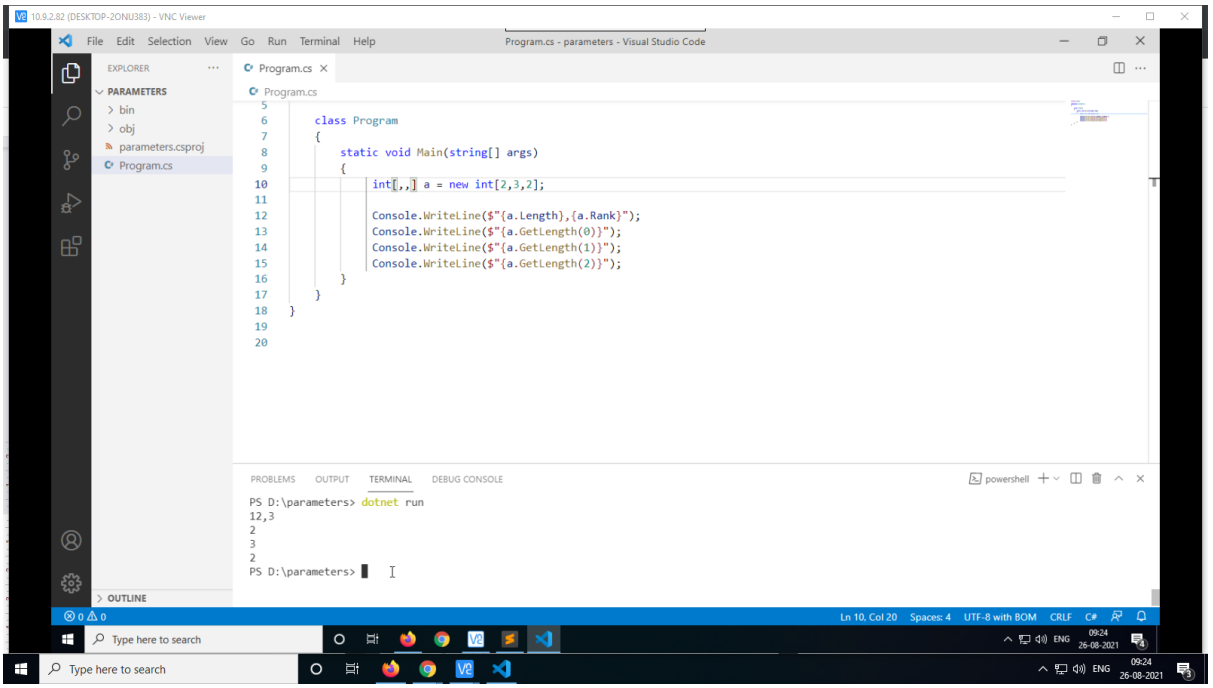

Program.cs - parameters - Visual Studio Code
PARAMETERS
bin
obj
parameters.csproj
Program.cs
static void Main(string[] args)
{
// int[,] a = new int[2,3] ;
// for (int i=0;i<2;i++)
// {
// for(int j=0;j<3;j++)
// {
// Console.Write("Enter value : ");
// a[i,j] = Convert.ToInt32(Console.ReadLine());
// }
// Console.WriteLine();
// }
int[] a = new int[3]{1,2,3};
foreach (int i in a)
{
Console.Write($"{i} ");
}
// for (int i=0;i<2;i++)
// {
// for(int j=0;j<3;j++)
// {
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
1 2 3
PS D:\parameters>
OUTLINE
Ln 25, Col 14 (133 selected) Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 09:08 26-08-2021


Program.cs - parameters - Visual Studio Code
PARAMETERS
bin
obj
parameters.csproj
Program.cs
static void Main(string[] args)
{
int[,] a = new int[2,3] ;
for (int i=0;i<2;i++)
{
for(int j=0;j<3;j++)
{
Console.Write("Enter value : ");
a[i,j] = Convert.ToInt32(Console.ReadLine());
}
Console.WriteLine();
}
foreach (int i in a)
{
Console.Write($"{i} ");
}
// for (int i=0;i<2;i++)
// {
// for(int j=0;j<3;j++)
// {
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
Enter value : 1
Enter value : 2
PS D:\parameters> dotnet run
Enter value : 1
Enter value : 2
Enter value : 3
OUTLINE
Ln 32, Col 34 Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 09:10 26-08-2021

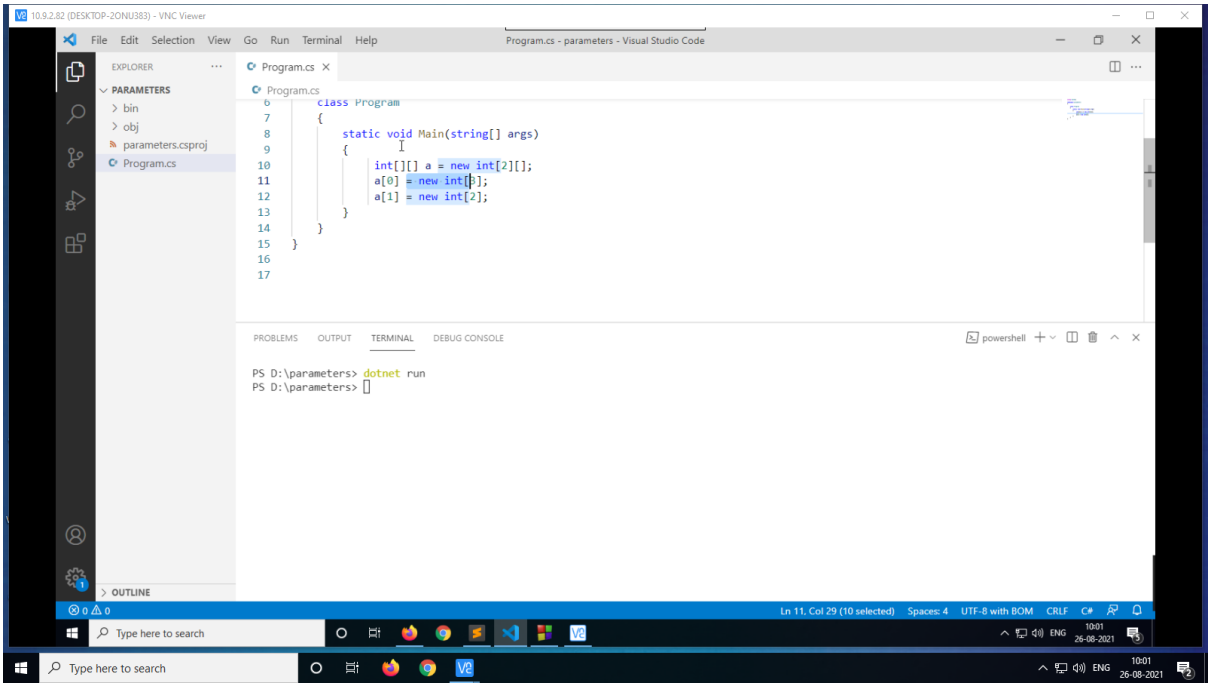

Program.cs - parameters - Visual Studio Code
PARAMETERS
bin
obj
parameters.csproj
Program.cs
class Program
{
static void Main(string[] args)
{
int[,] a = new int[2,3] {
{1,2,3},
{2,3,4}
} ;
Console.WriteLine($"{a.Length},{a.Rank}");
}
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS D:\parameters> dotnet run
6,2
PS D:\parameters>
Ln 15, Col 1 Spaces: 4 UTF-8 with BOM CRLF C#

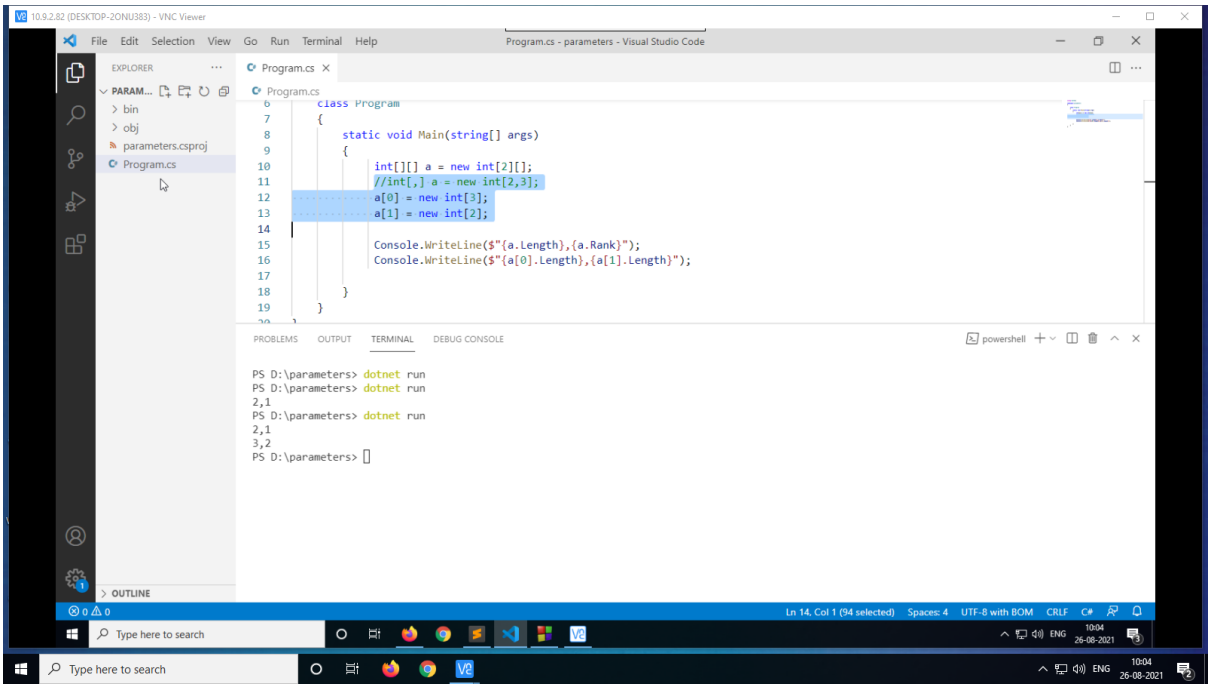

Program.cs - parameters - Visual Studio Code
PARAMETERS
bin
obj
parameters.csproj
Program.cs
class Program
{
static void Main(string[] args)
{
int[,] a = new int[2,3] {
{1,2,3},
{2,3,4}
} ;
Console.WriteLine($"{a.Length},{a.Rank}");
Console.WriteLine($"{a.GetLength(0)}");
Console.WriteLine($"{a.GetLength(1)}");
}
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS D:\parameters> dotnet run
6,2
2
3
PS D:\parameters>
Ln 16, Col 48 Spaces: 4 UTF-8 with BOM CRLF C#

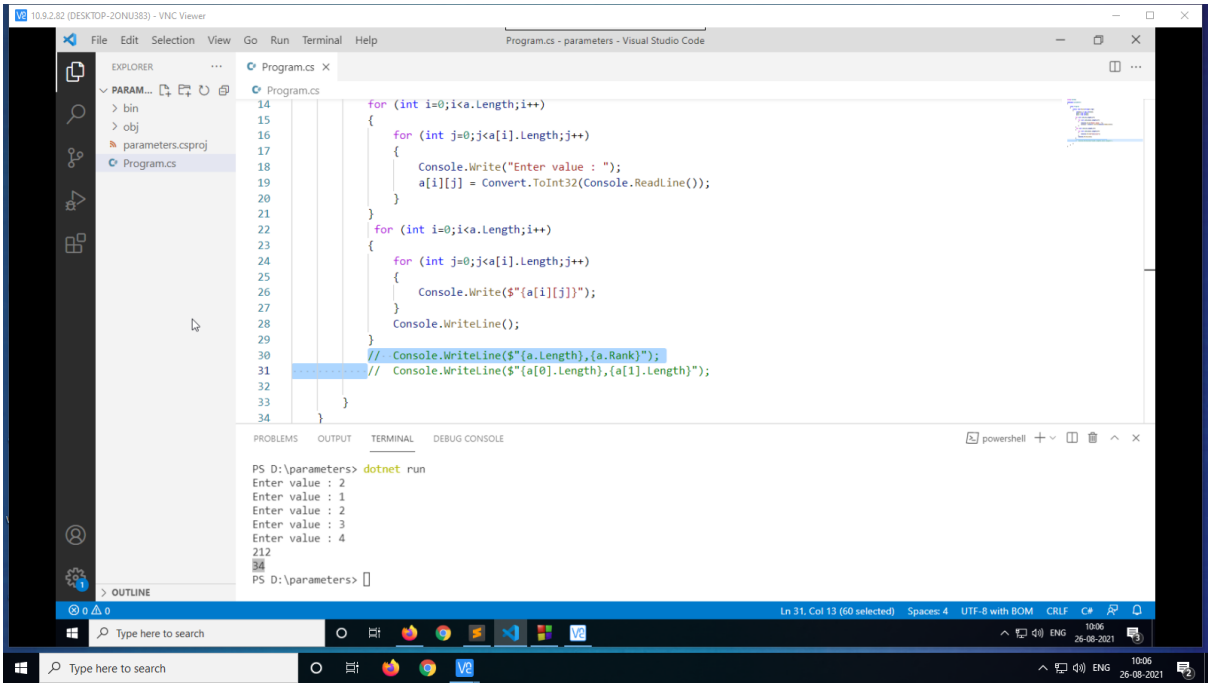


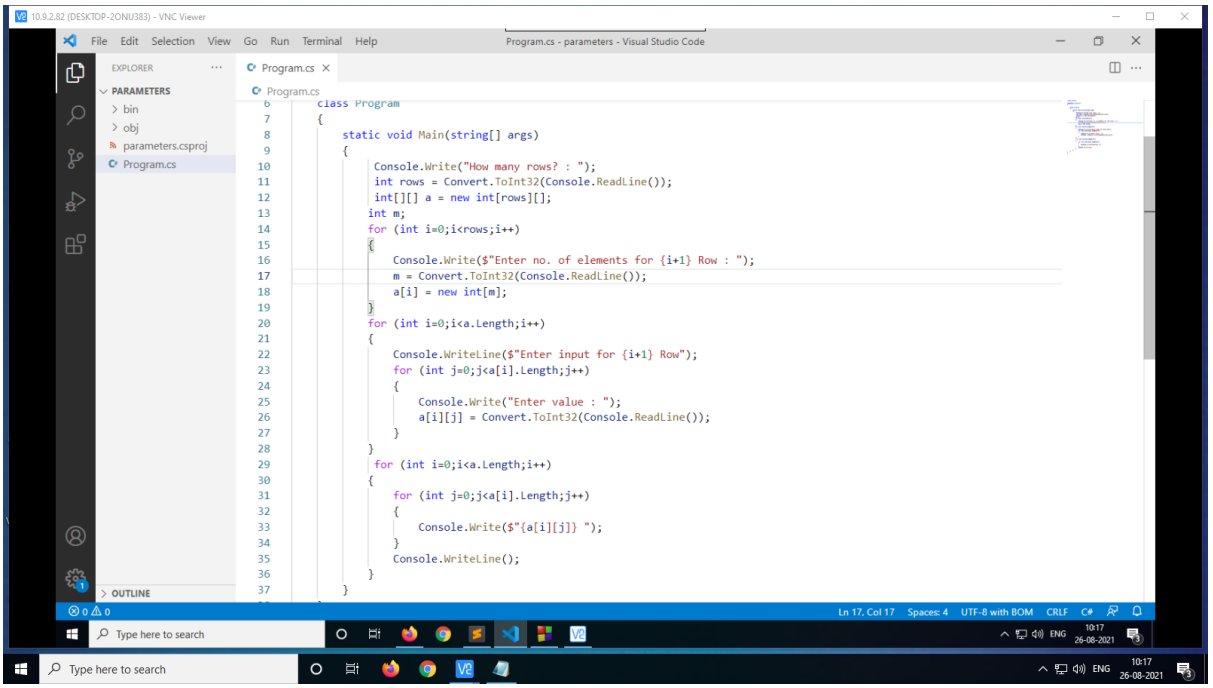


Jagged Array


10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAMETERS
bin
obj
parameters.csproj
Program.cs
Program.cs
class Program
{
static void Main(string[] args)
{
int[][] a = new int[2][];
a[0] = new int[3];
a[1] = new int[2];
}
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
PS D:\parameters>
OUTLINE
Ln 11, Col 29 (10 selected) Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 10:01 26-08-2021


10.9.2.82 (DESKTOP-2ONU383) - VNC Viewer
File Edit Selection View Go Run Terminal Help
Program.cs - parameters - Visual Studio Code
EXPLORER
PARAM...
bin
obj
parameters.csproj
Program.cs
Program.cs
class Program
{
static void Main(string[] args)
{
int[][] a = new int[2][];
//int[,] a = new int[2,3];
a[0] = new int[3];
a[1] = new int[2];
Console.WriteLine($"{a.Length},{a.Rank}");
Console.WriteLine($"{a[0].Length},{a[1].Length}");
}
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
PS D:\parameters> dotnet run
PS D:\parameters> dotnet run
2,1
PS D:\parameters> dotnet run
2,1
3,2
PS D:\parameters>
OUTLINE
Ln 14, Col 1 (94 selected) Spaces: 4 UTF-8 with BOM CRLF C#
Type here to search
ENG 10:04 26-08-2021





Write a C# Program to create 3x4 int array Get the nos from user and print the same.

Write a C# Program to create a jagged array of size 3 get the no of elements for each subarray from user and take the values from user and print the same