

C# .NET

C# Introduction

- It Was developed by the Microsoft Corporation.
- It is high level programming language.
- It is an object oriented programming language.
- It is case sensitive programming language.
- It extension .cs.
- It is a very easy programming language.
- It belongs to C family.
- It is used to develop various types applications.
- Console,Windows,Web and Mobile applications.

C# Features

- Implicitly typed local variables
- Object and collection initializers
- Auto-Implemented properties
- Anonymous types
- Extension methods
- Query expressions
- Lambda expressions
- Expression trees
- Partial Methods

C# Program Structure

Predefined Namespace

User defined Namespace

Class Declaration

Data types

Methods.

Structure Definitions

Namespace

It is set of classes, interfaces, enumerations, delegates etc....

Class

It is the collection of datatypes and methods.

C# Program Structure

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

→ Predefined Namespaces

```
namespace Test {
```

→ User defined Namespace

```
class Program
```

```
{
```

→ Class Declaration

```
static void Main(string[] args)
```

```
{
```

→ Method

```
}
```

Document Area

```
}
```

```
}
```

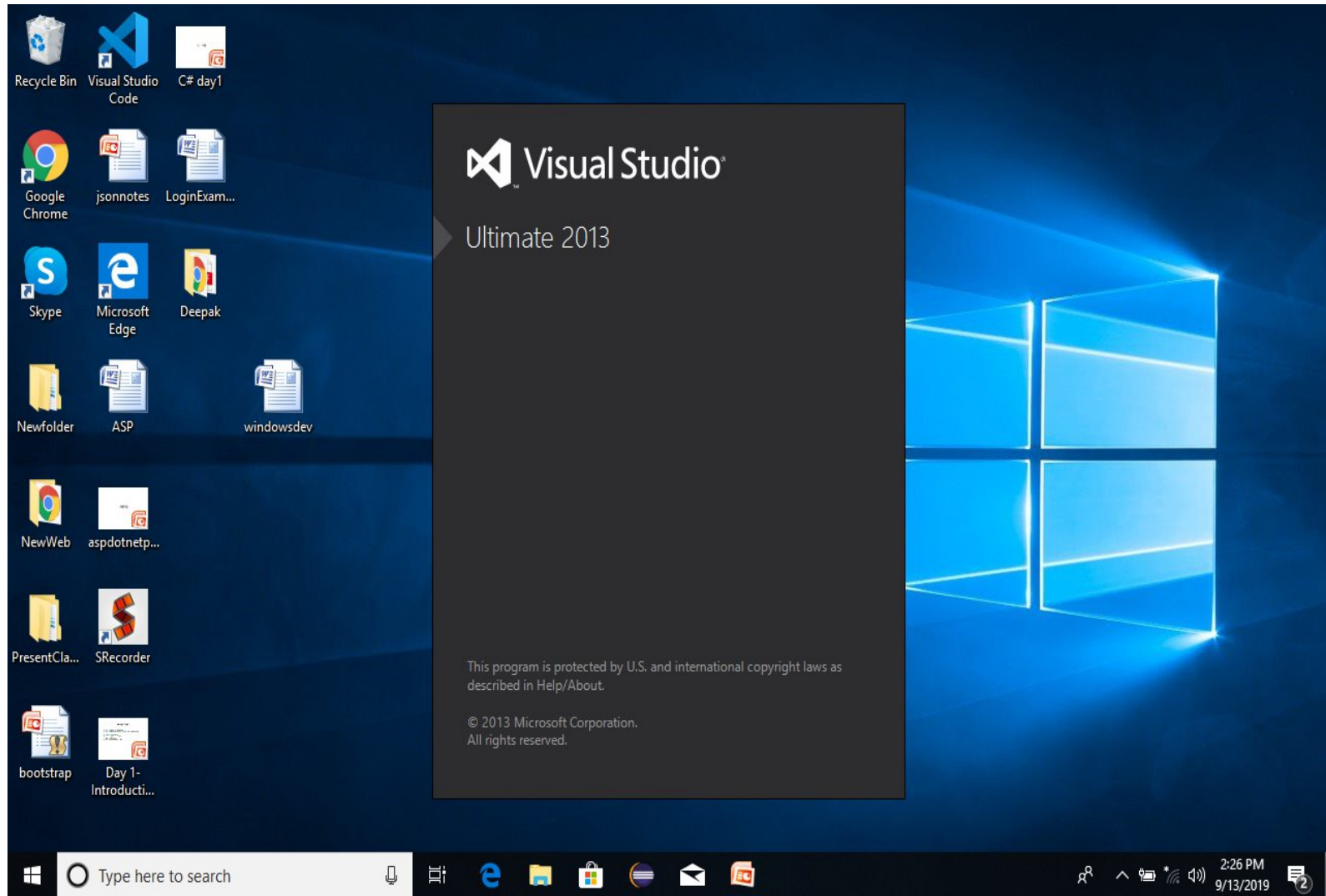
How to develop a program in C#?

1. Open the project
2. Code the program using C#.
3. Execute using Ctrl+F5.
4. These Steps will be repeated for all the programs.

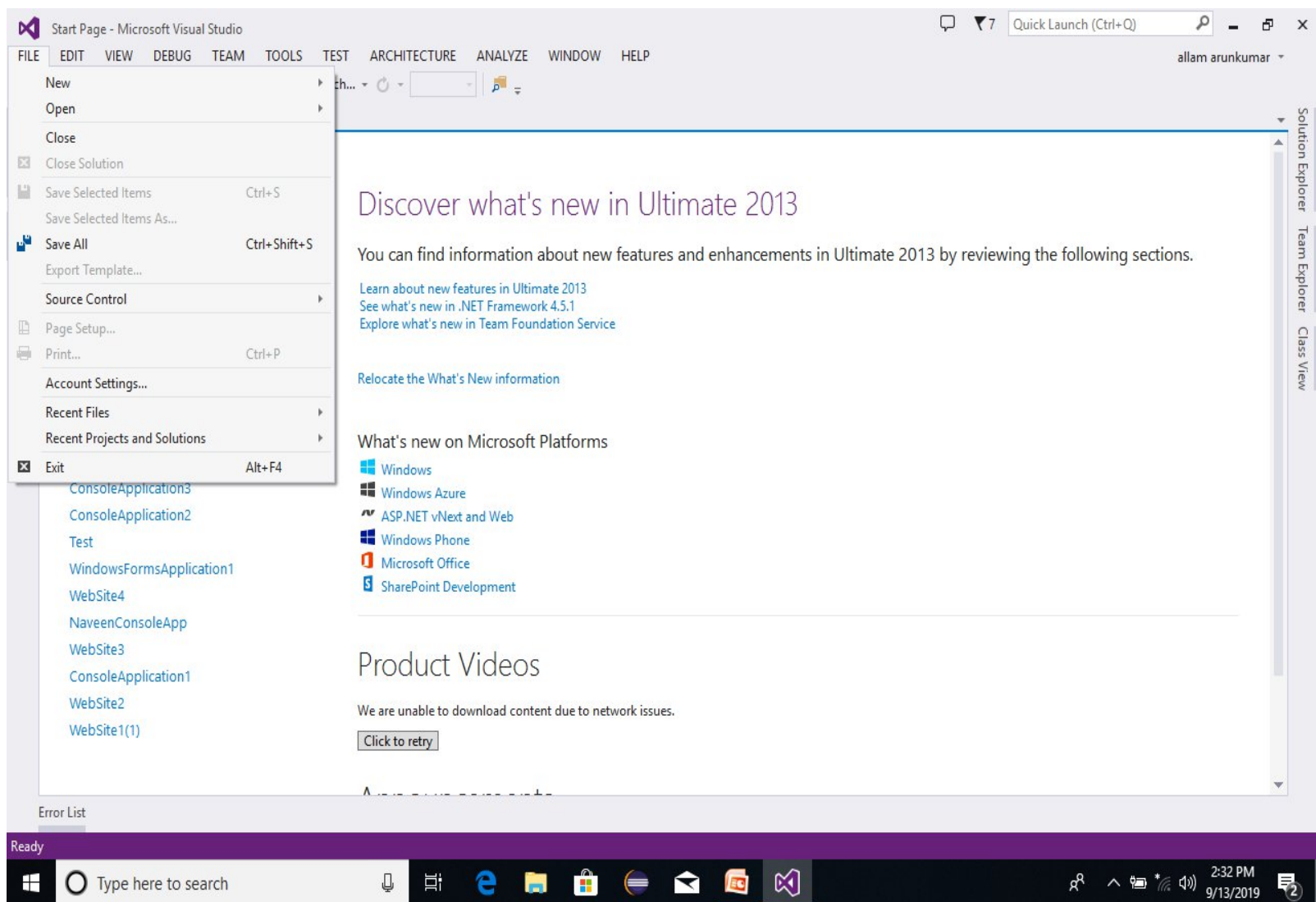
Visual Studio Introduction

1. It is an IDE(Integrated Development Environment).
2. It is used develop software , web and Mobile applications.
3. It is Introduced by Microsoft.
4. In order to work with any project in Visual Studio tool we need know about Solution Explorer and Server Explorer .
5. Solution Explorer in which project and its respect files can be added.
6. Sever Explorer in which Databases can deal the with the DataBases.

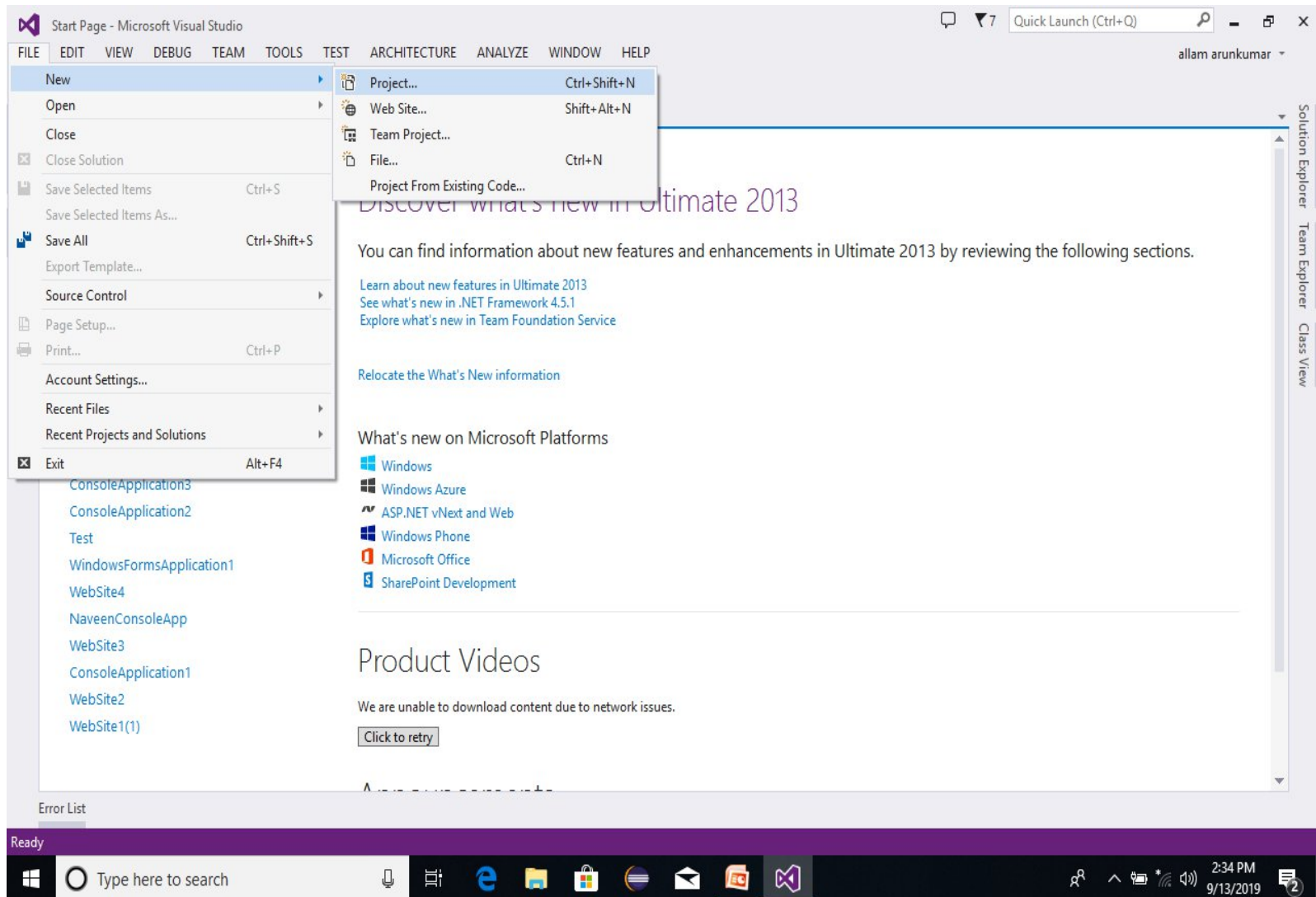
Creating new project in VS



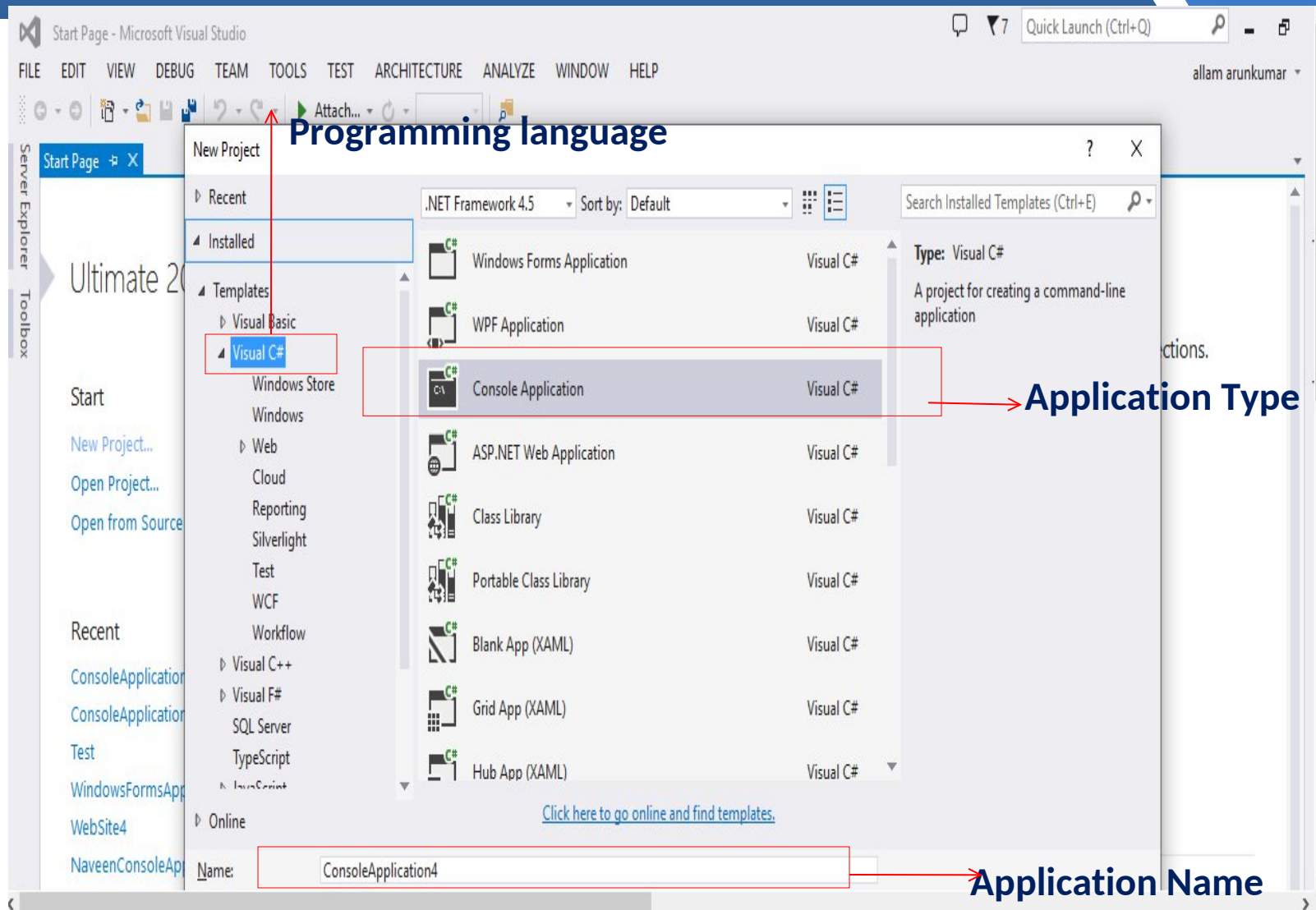
Creating new project in VS



Creating new project in VS



Creating new project in VS



Coding new program in VS

The screenshot shows the Microsoft Visual Studio IDE with a new console application named 'ConsoleApplication4'. The main code file, 'Program.cs', is open and displays the following C# code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication4
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            |
        }
    }
}
```

The code is written in a standard C# style with blue keywords and black text. The 'Main' method is currently empty, with the cursor positioned at the start of the line. The IDE interface includes the menu bar (File, Edit, View, Project, Build, Debug, Team, Tools, Test, Architecture, Analyze, Window, Help), the toolbar, and the Solution Explorer on the right side. The Solution Explorer shows the project structure with 'Program.cs' selected. The status bar at the bottom indicates the current line (Ln 13), column (Col 13), and character (Ch 13). The Windows taskbar is visible at the very bottom, showing the Start button, search bar, and various application icons.

Coding new program in VS

ConsoleApplication4 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

allam arunkumar

Program.cs* [X]

ConsoleApplication4.Program

Main(string[] args)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication4
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
        }
    }
}
```

177 %

Error List

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'ConsoleApplication4' (1 project)

- ConsoleApplication4
 - Properties
 - References
 - App.config
 - Program.cs

Solution Explorer in which, we Add Files to the project, all the files which are present project will be seen here

Document Area where work here

Ready

Ln 13 Col 13 Ch 13 INS

Type here to search

2:55 PM 9/13/2019

Coding new program in VS

ConsoleApplication4 - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

allam arunkumar

Program.cs

ConsoleApplication4.Program

Main(string[] args)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication4
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

After writing the code

press CTRL + F5
To
Execute the Program

177 %

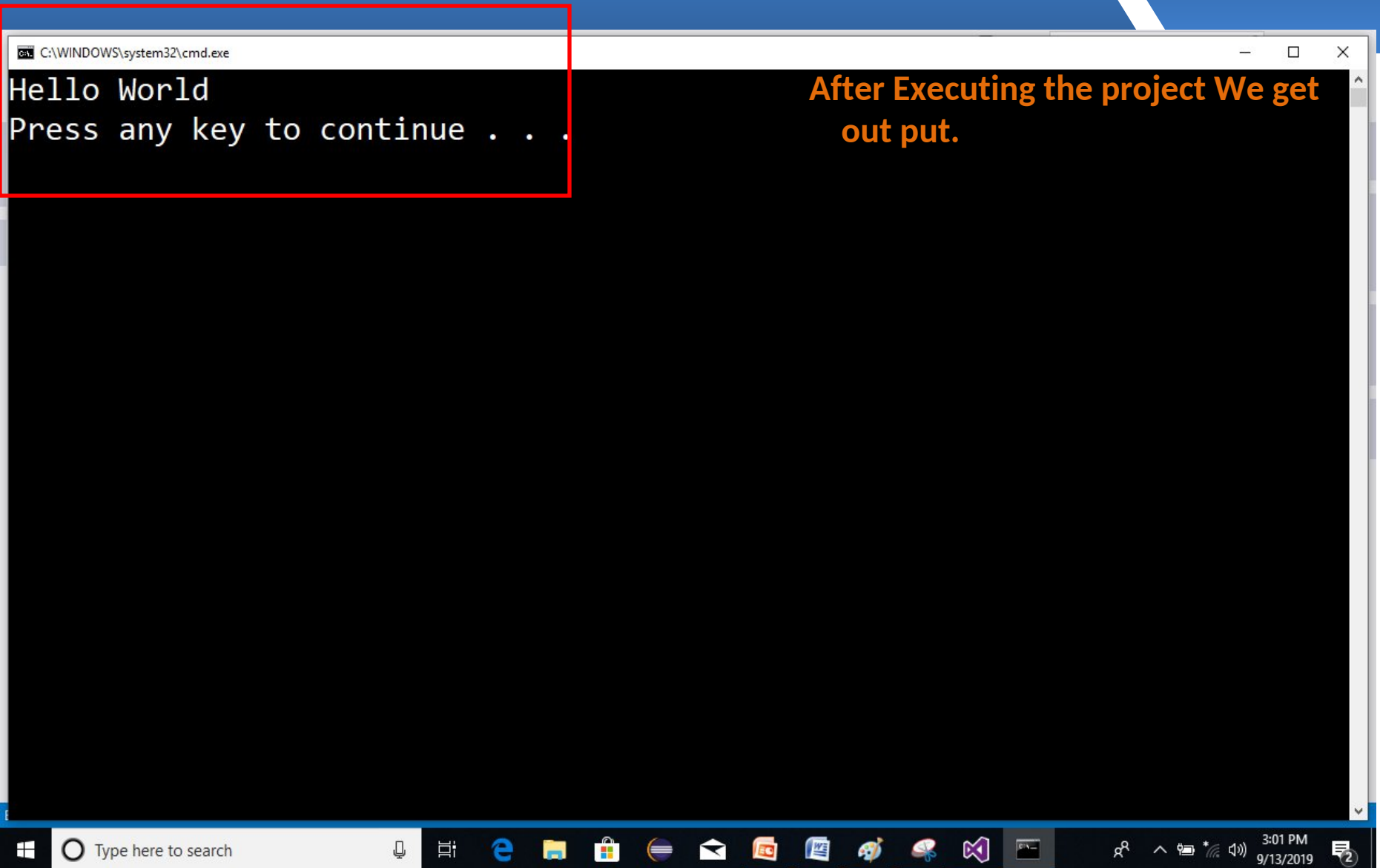
Error List

Ready Ln 13 Col 44 Ch 44 INS

Type here to search

2:58 PM 9/13/2019

Coding new program in VS



Data Types in C#

- char 1 byte
- int 4 byte
- float 4 byte
- double 8 byte
- decimal 16 byte
- String
- class

Input and Out Put Methods

Out put Methods

These methods can be used to display the statements and values inside the console App.

They are

`Console.WriteLine()`

`Console.Write()`

Input and Out Put Methods

Input Methods

These are used to read values from the console.

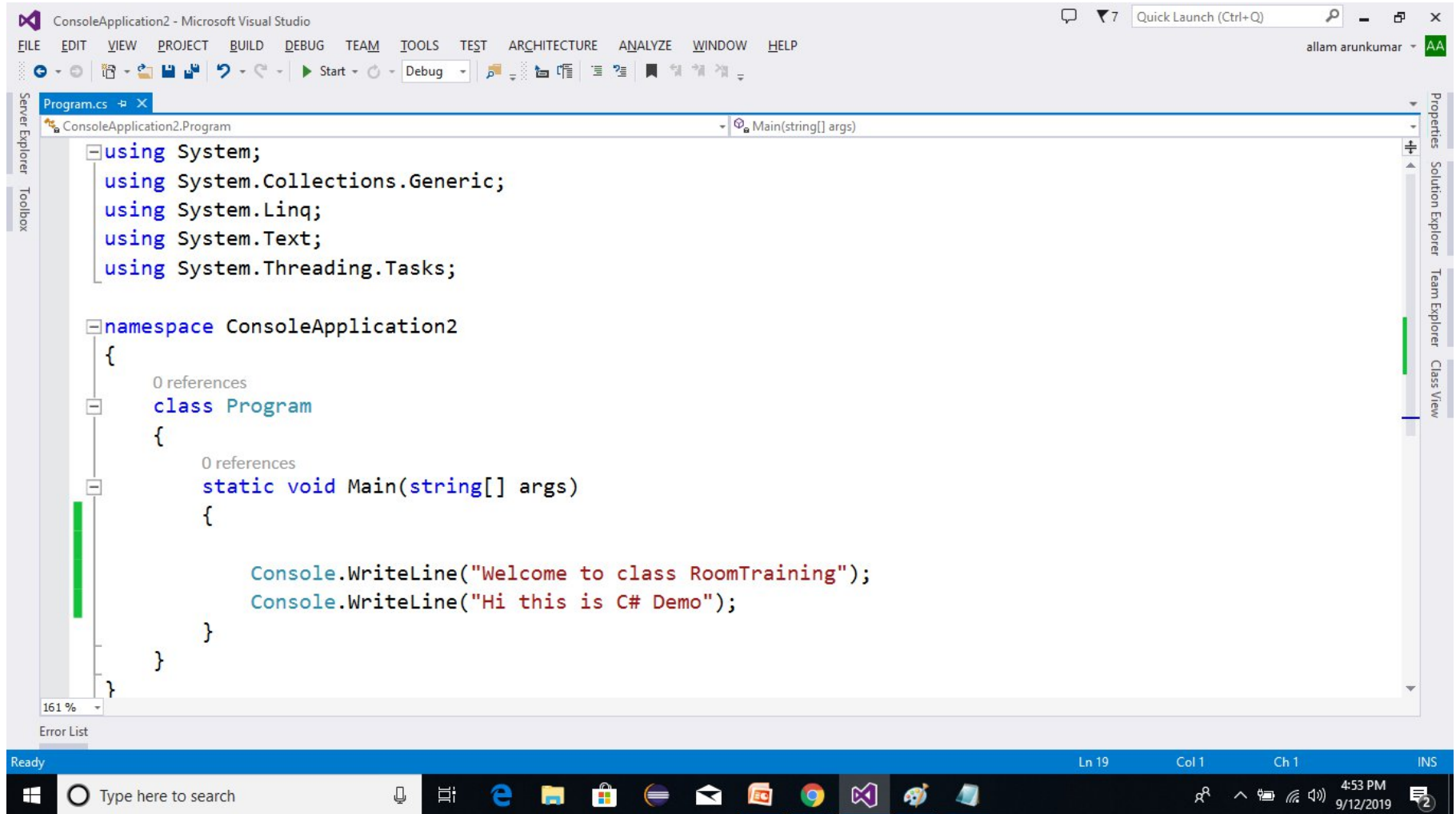
They are

`Console.ReadLine()`

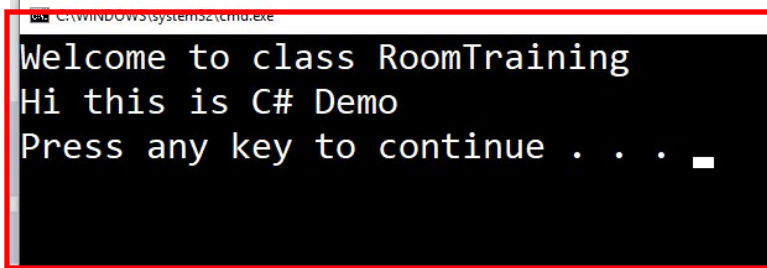
`Console.Read()`

`Console.ReadKey()`

Example on out put methods



Coding new program in VS

A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.exe'. The window contains the following text: 'Welcome to class RoomTraining', 'Hi this is C# Demo', and 'Press any key to continue . . .'. A red rectangular box highlights the first three lines of text.

```
C:\WINDOWS\system32\cmd.exe
Welcome to class RoomTraining
Hi this is C# Demo
Press any key to continue . . .
```

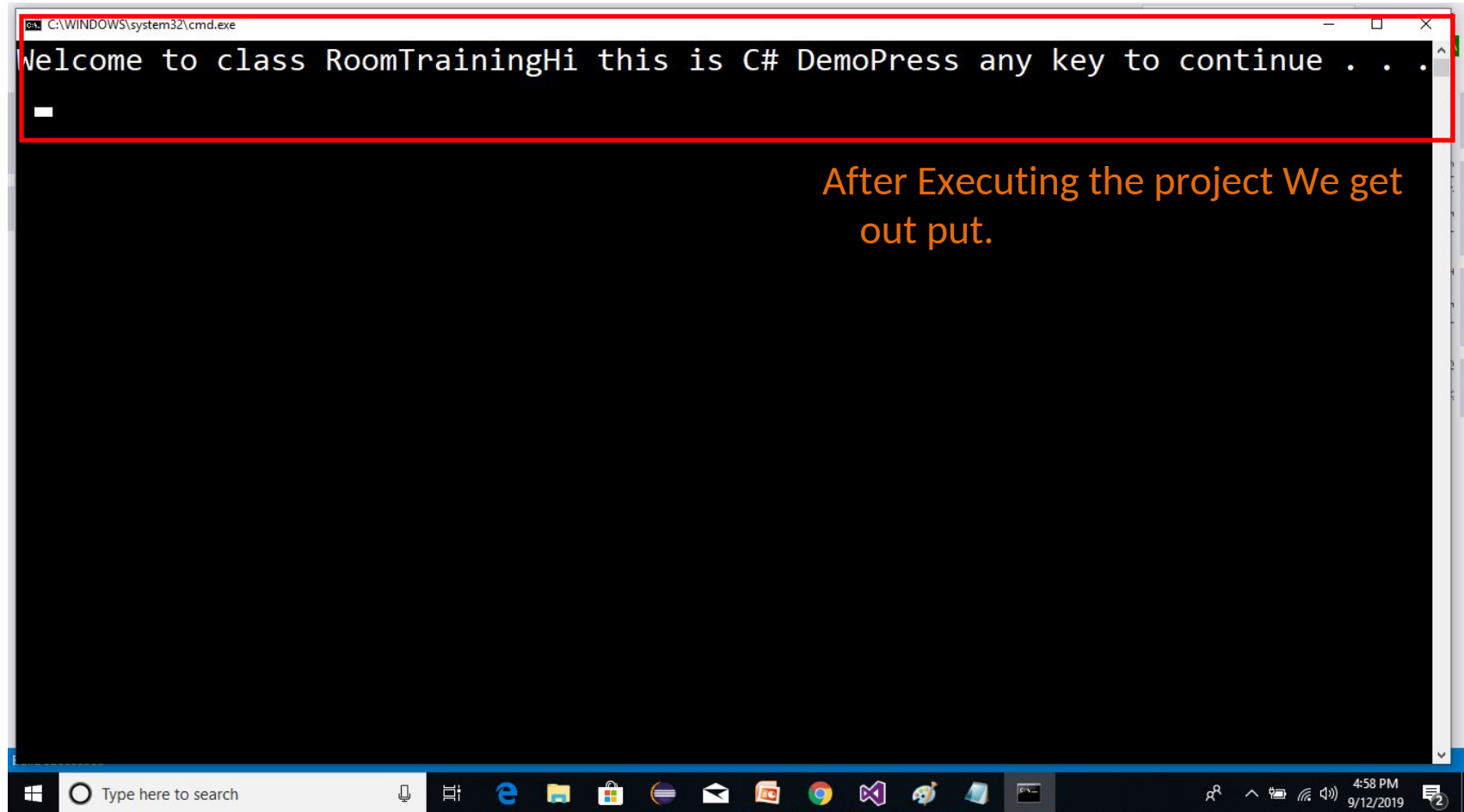
After Executing the project We get
out put.

Example on out put methods

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to class RoomTraining");
            Console.WriteLine("Hi this is C# Demo");
        }
    }
}
```

After Executing VS



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINDOWS\system32\cmd.exe". The command prompt displays the text "Welcome to class RoomTrainingHi this is C# DemoPress any key to continue . . ." followed by a cursor. A red rectangular box highlights the text and the cursor. To the right of the command prompt, the text "After Executing the project We get out put." is written in orange. The Windows taskbar is visible at the bottom, showing the search bar, task view button, and several application icons. The system tray on the right shows the date and time as "4:58 PM 9/12/2019".

```
C:\WINDOWS\system32\cmd.exe
Welcome to class RoomTrainingHi this is C# DemoPress any key to continue . . .
_
```

After Executing the project We get out put.

Example on Data Types with Out put methods

```
static void Main(string[] args)
{
    int a = 10;
    float x = 4.5f;
    double d = 7.8;
    string s = "Welcome";
    decimal m = 5.6m;
    Console.WriteLine("{0} {1} {2} {3} {4}",a,x,d,s,m);
}
```


Input and Out Put Methods

- Input Methods
 - These are used to read values from the console.
 - They are
 - `Console.ReadLine()`
 - `Console.Read()`
 - `Console.ReadKey()`

Input and Out Put Methods

- `Console.ReadLine()`
It is used to read a string
- `Int.Parse(Console.ReadLine())`
It is used to Read Integer Value
- `Float.Parse(Console.ReadLine())`
It is used to Read Float value.
- `Double.Parse(Console.ReadLine())`
It is used to Read double value.
- `Bool.Parse(Console.ReadLine())`
It is used to boolean value
- `Console.Read()`
It used to Read ASCII Value of the given Character.
- `Console.ReadKey()`
It is used to read a key from the console

Reading values

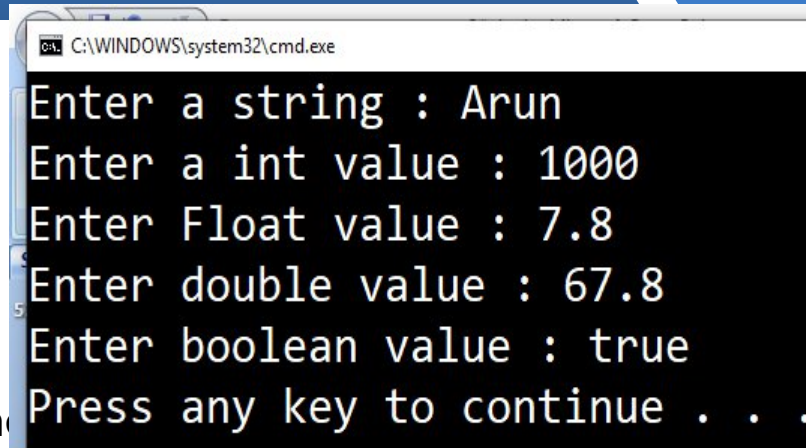
```
static void Main(string[] args)
{
    Console.Write("Enter a string : ");
    string s = Console.ReadLine();

    Console.Write("Enter a int value : ");
    int a = int.Parse(Console.ReadLine());
```

```
    Console.Write("Enter Float value : ");
    float f = float.Parse(Console.ReadLine());
```

```
    Console.Write("Enter double value : ");
    double d = double.parse(Console.ReadLine());
```

```
    Console.Write("Enter boolean value : ");
    bool b = bool.Parse(Console.ReadLine());
}
```



```
C:\WINDOWS\system32\cmd.exe
Enter a string : Arun
Enter a int value : 1000
Enter Float value : 7.8
Enter double value : 67.8
Enter boolean value : true
Press any key to continue . . .
```

Operators In C#

Operators in C#

- Arithmetic operators .
- Relational operators.
- Logical Operators.
- Assignment operators.
- conditional operators.
- Increment and Decrement operators.
- Bitwise Operators.// home work

Arithmetic operators .

- + -> add
- - -> sub
- * -> mul
- / -> div-->quo $5/2 \text{ ---> } 2$
- % -> mode->Rem $5\%2 \text{ ---> } 1$

Arithmetic operators .

```
static void Main(string[] args)
{
    int a = 5;
    int b = 2;
    Console.WriteLine("Sum:{0}",a+b);
    Console.WriteLine("Sub:{0}",a-b);
    Console.WriteLine("Mul:{0}",a*b);
    Console.WriteLine("Quo:{0}",a/b);
    Console.WriteLine("Rem:{0}",a%b);
}
```

Sum : 7

Sub : 3

Mul : 10

Quo : 2

Rem: 1

Relational operators:

- > ->Greter
- < ->less
- >= ->Greaterthan or equal
- <= ->Lesserthan or equal
- == ->(comparision)
- != ->(Not Equal)

Relational operators

```
static void Main(string[] args)
```

```
{
```

```
    int a = 5;
```

```
    int b = 2;
```

```
    Console.WriteLine("a > b : {0}", a > b);
```

```
    Console.WriteLine("a >= b : {0}", a >= b);
```

```
    Console.WriteLine("a < b : {0}", a < b);
```

```
    Console.WriteLine("a <= b : {0}", a <= b);
```

```
    Console.WriteLine("a == b : {0}", a == b);
```

```
    Console.WriteLine("a != b : {0}", a != b);
```

```
}
```

a > b:True

a >= b :True

a < b : False

a <= b : False

a == b : False

a != b :True

Logical Operators

&&- --> and

|| ---> or

! ---> not

con1	con2	con1 && con2
------	------	--------------

true	true	true
------	------	------

true	false	false
------	-------	-------

false	true	false
-------	------	-------

false	false	false
-------	-------	-------

con1	con2	con1 con2
------	------	--------------

true	true	true
------	------	------

true	false	true
------	-------	------

false	true	true
-------	------	------

false	false	false
-------	-------	-------

con	!con
-----	------

true	false
------	-------

false	true
-------	------

Assignment operators

=	Assignment operator
+=	Additive Assignment
-=	Subtractive Assignment
*=	Multiplicative Assignment
/=	Division Assignment

Assignment Operators program

```
static void Main(string[] args)
```

```
{
```

```
    int a = 10;
```

```
    int b = 20;
```

```
    Console.WriteLine("a = {0}", a);
```

```
    a += b;        // a = a+b
```

```
    Console.WriteLine("a = {0}", a);
```

```
    a -= b;
```

```
    Console.WriteLine("a = {0}", a);
```

```
    a *= b;
```

```
    Console.WriteLine("a = {0}", a);
```

```
    a /= b;
```

```
    Console.WriteLine("a = {0}", a);
```

```
}
```

a = 10

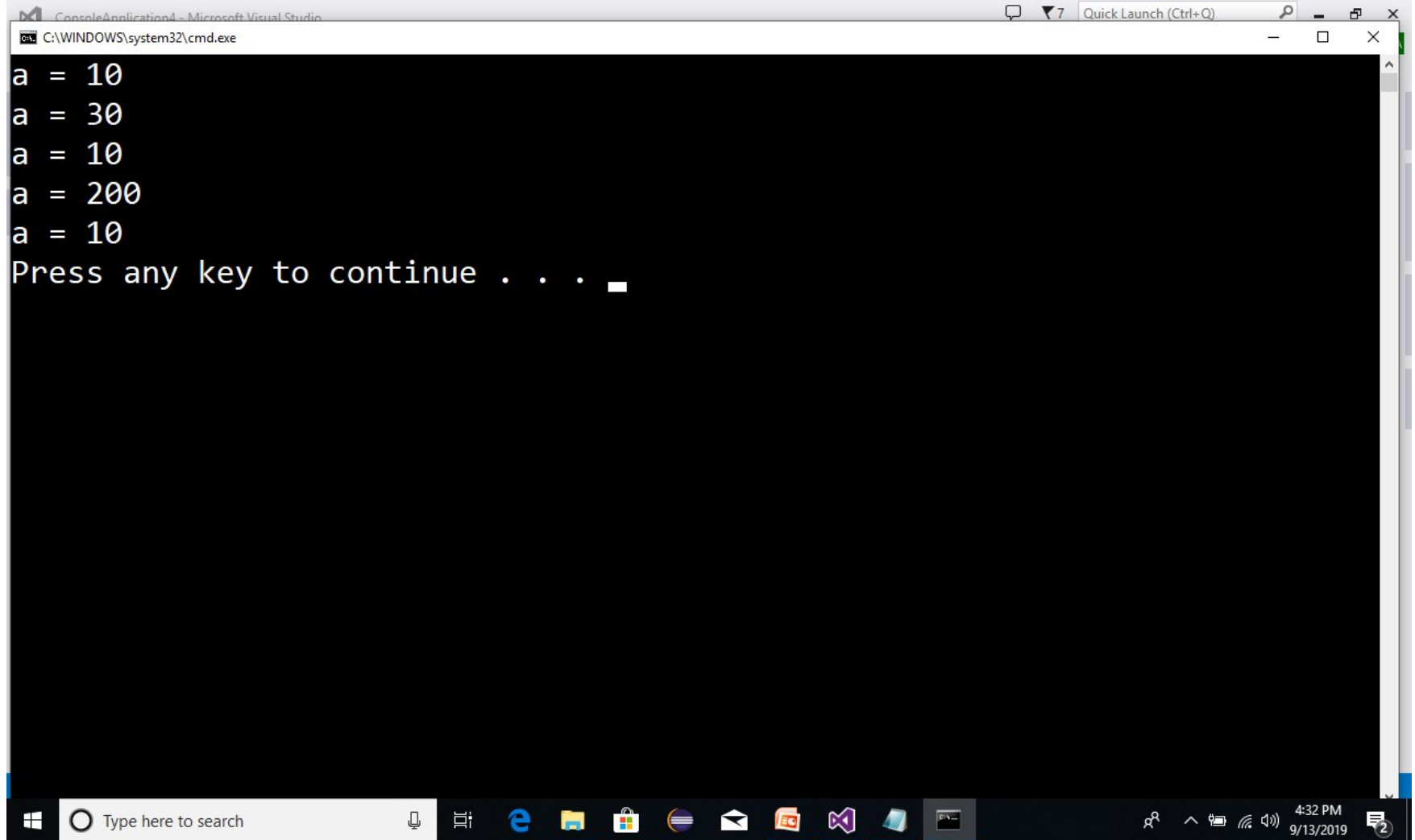
a = 30

a = 10

a = 200

a = 10

Assignment operators:



The image shows a screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window is open within a Microsoft Visual Studio environment, as indicated by the "ConsoleApplication4 - Microsoft Visual Studio" title bar. The command prompt displays the following text:

```
a = 10  
a = 30  
a = 10  
a = 200  
a = 10  
Press any key to continue . . .
```

The window has a standard Windows taskbar at the bottom. The taskbar includes the Start button, a search bar with the text "Type here to search", and several application icons: File Explorer, Microsoft Store, Edge, Mail, and Visual Studio. The system tray on the right shows the time as 4:32 PM on 9/13/2019, along with icons for network, volume, and power.

Conditional operators

? : (Ternary Operator)

con?stmt1:stmt2;

con-->True--->stmt1 can be executed

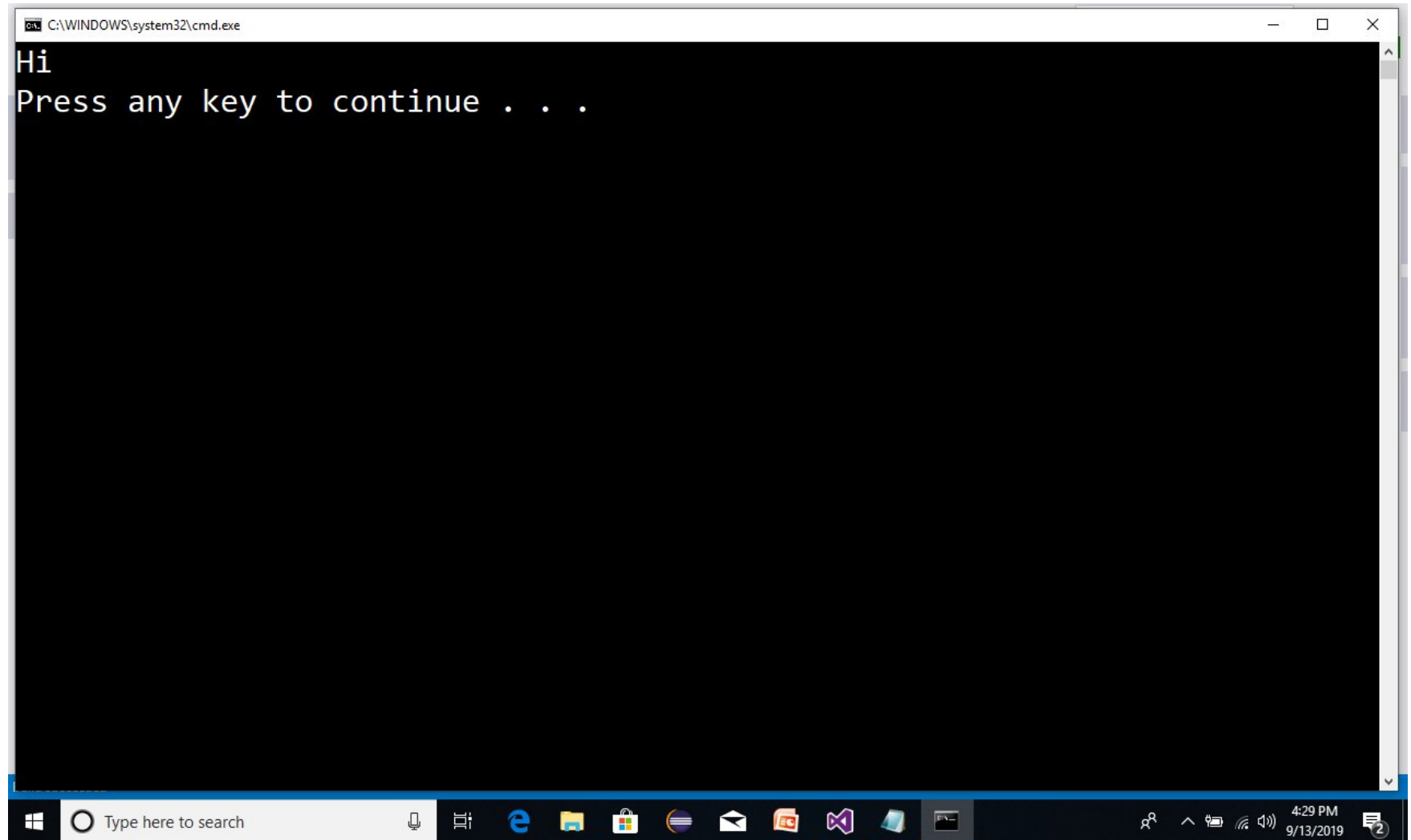
con--->False-->stmt2 can be executed

Conditional Operators program

```
static void Main(string[] args)
{
    bool con = true;
    Console.WriteLine(con?"Hi":"Bye");
}
```

Hi

Conditional operators:

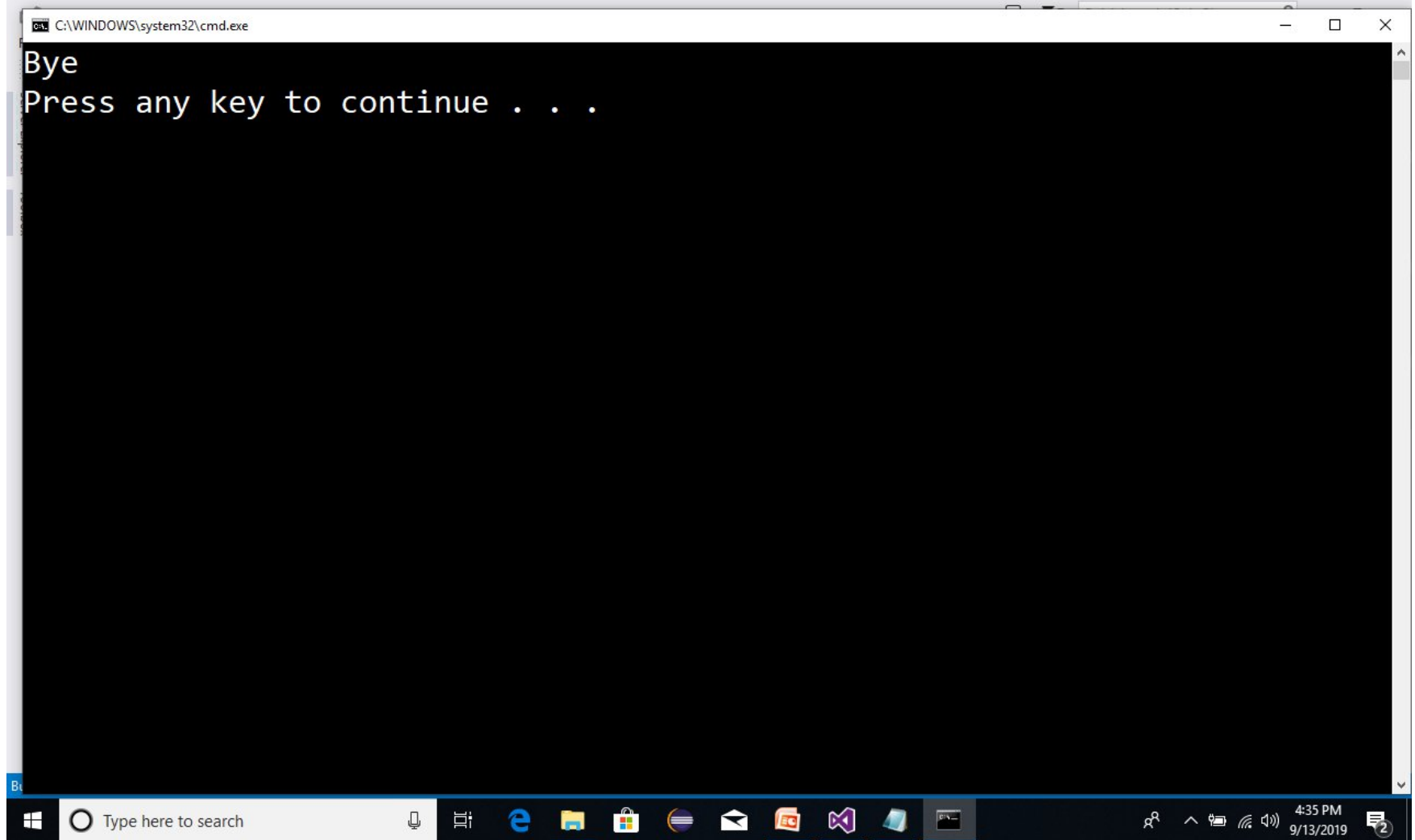


Conditional Operators program

```
static void Main(string[] args)
{
    bool con = false;
    Console.WriteLine(con?"Hi":"Bye");
}
```

Bye

Conditional operators:



Increment and Decrement operators

- ++ Increment Operator
- Decrement Operator

Increment and Decrement Operators program

```
static void Main(string[] args)
{
    int a = 10;
    int b = 20;
    Console.WriteLine("a = {0}",a++); // Post Increment
    Console.WriteLine("a = {0}", a);
    Console.WriteLine("b = {0}", ++b);
    Console.WriteLine("b = {0}", b);
}
```

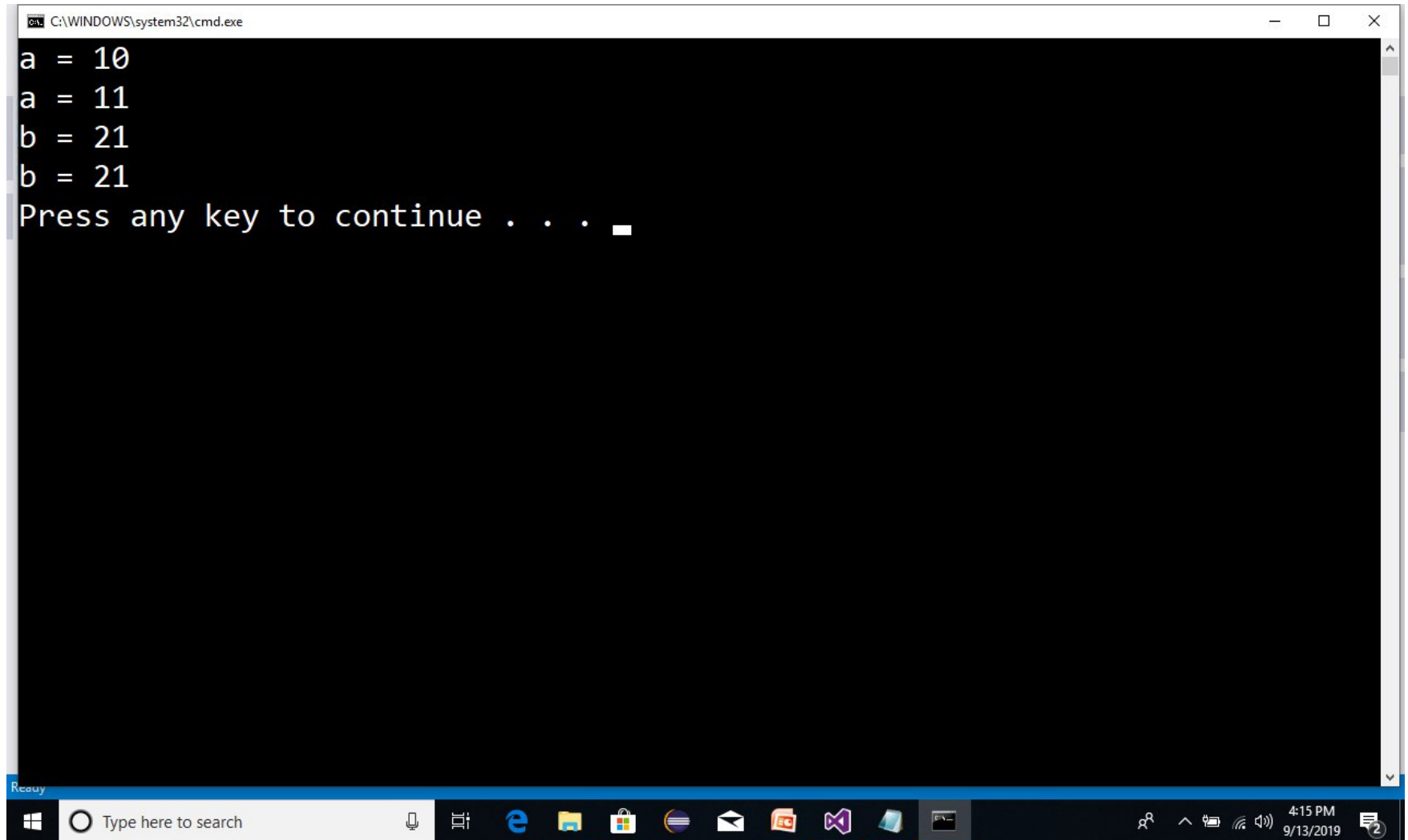
a = 10

a = 11 ent

a = 21

a = 21

Increment decrement operators:



The image shows a Windows Command Prompt window with the title bar "C:\WINDOWS\system32\cmd.exe". The window contains the following text:

```
a = 10  
a = 11  
b = 21  
b = 21  
Press any key to continue . . .
```

The window is running on a Windows 10 desktop. The taskbar at the bottom shows the Start button, a search bar with the text "Type here to search", and several pinned application icons including File Explorer, Microsoft Store, Edge, Mail, and others. The system tray on the right shows the time as 4:15 PM on 9/13/2019.