

# Object Oriented Programming

## Lab Manual 2

### Introduction

In this course, we will learn a new programming language that is known as C# (C Sharp). Now that you have successfully installed and set up the Visual Studio on your computer, let's learn about coding some basic operations in C#.

**Let's do some coding.**

### Output Operation in C#

We have learned in the previous manual about printing output on the console screen. For revision, let's use the “Hello World” example.

#### “Console.Write(“Hello World”)”

This line will print the message inside the quotation mark on the existing line where the cursor is in the console.

**Task:** To understand this concept, try writing a program that prints “Hello World” twice on the screen.

#### **Solution:**

Write the following code into the main function of the code and execute the program by clicking on the start button.

Code:	Output:
<pre>1 using System; 2 using System.Collections.Generic; 3 using System.Linq; 4 using System.Text; 5 using System.Threading.Tasks; 6 7 namespace Test 8 { 9     class Program 10    { 11        static void Main(string[] args) 12        { 13            Console.Write("HELLO WORLD!!"); 14            Console.Write("HELLO WORLD!!"); 15            Console.ReadKey(); 16        } 17    } 18 } 19</pre>	<pre>C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe HELLO WORLD!!HELLO WORLD!!</pre>

# Object Oriented Programming

## Lab Manual 2

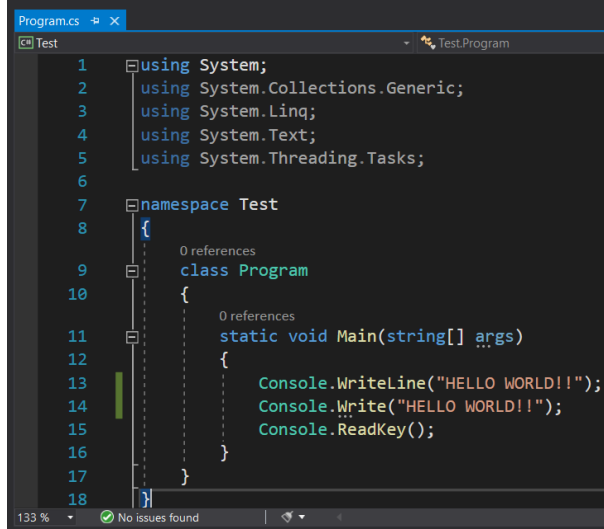
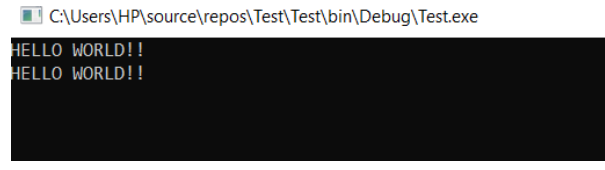
**“Console.WriteLine(“Hello World”)”**

This line will print the message on the next line.

**Task:** To understand this concept, try writing a program that prints “Hello World” on two separate lines on the screen.

**Solution:**

Write the following code into the main function of the code and execute the program by clicking on the start button.

Code:	Output:
	

## Variables in C#

Just like other programming languages, we have variables of various data types that are used in the programs according to user requirements.

# Object Oriented Programming

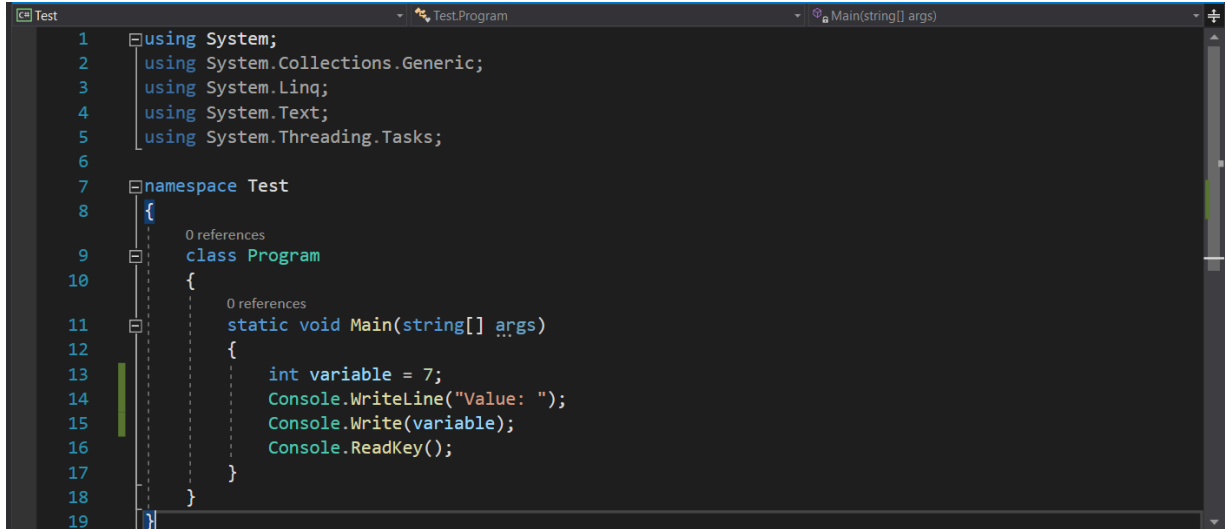
## Lab Manual 2

### Integer Datatype

The declaration and use of integer datatype are very simple and straightforward.

Look at the following code snippet to understand the use of the “**int**” datatype in C#.

#### Code:



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Test
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             int variable = 7;
14             Console.WriteLine("Value: ");
15             Console.Write(variable);
16             Console.ReadKey();
17         }
18     }
19 }
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
Value:
7
```

# Object Oriented Programming

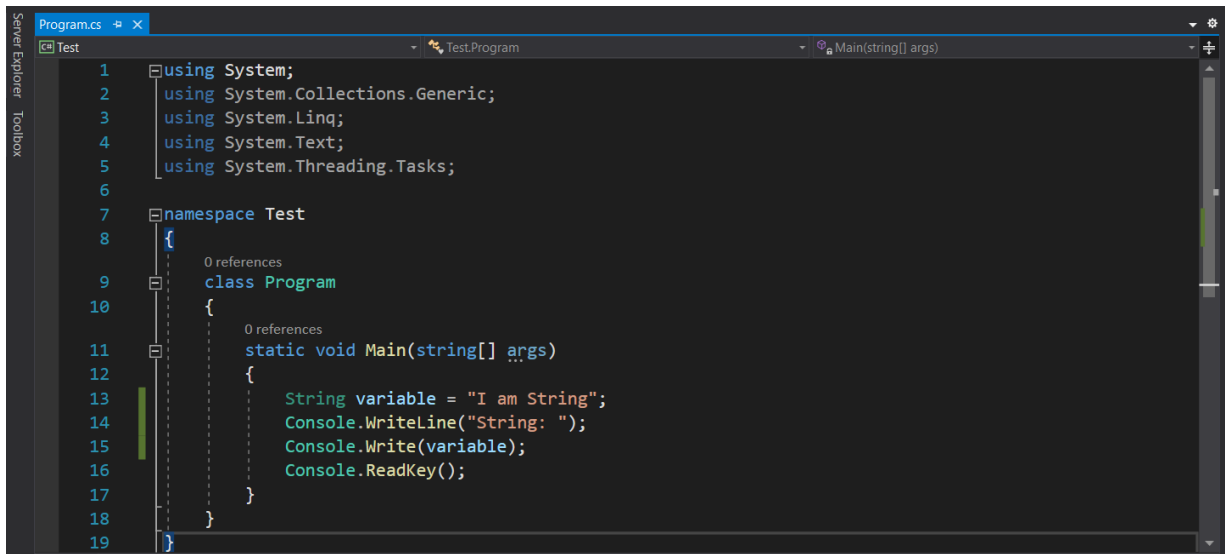
## Lab Manual 2

### String Datatype

Similarly, the string datatype is very simple to use. The “**String**” keyword is used for declaring a string-type keyword.

Look at the following code snippet to understand the use of the “**String**” datatype in C#.

#### Code:



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Test
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            String variable = "I am String";
14            Console.WriteLine("String: ");
15            Console.Write(variable);
16            Console.ReadKey();
17        }
18    }
19 }
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
String:
I am String
```

# Object Oriented Programming

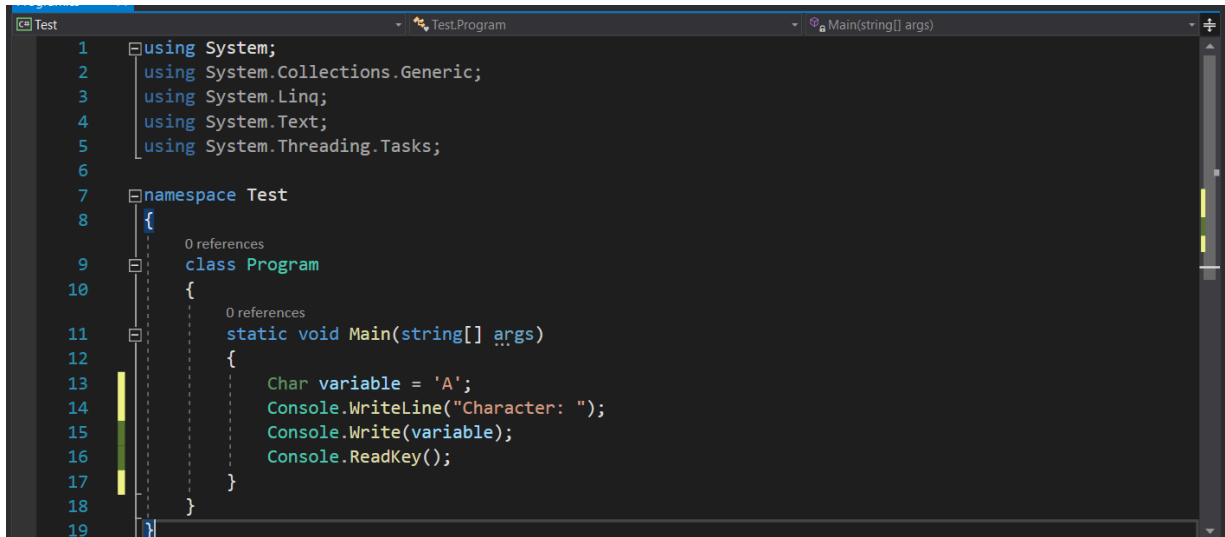
## Lab Manual 2

### Character Datatype

Variable of character datatype are declared using the “**Char**” keyword.

Look at the following code snippet to understand the use of the “**Char**” datatype in C#.

#### Code:



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Test
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            Char variable = 'A';
16            Console.WriteLine("Character: ");
17            Console.Write(variable);
18            Console.ReadKey();
19        }
20    }
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
Character:
A
```

# Object Oriented Programming

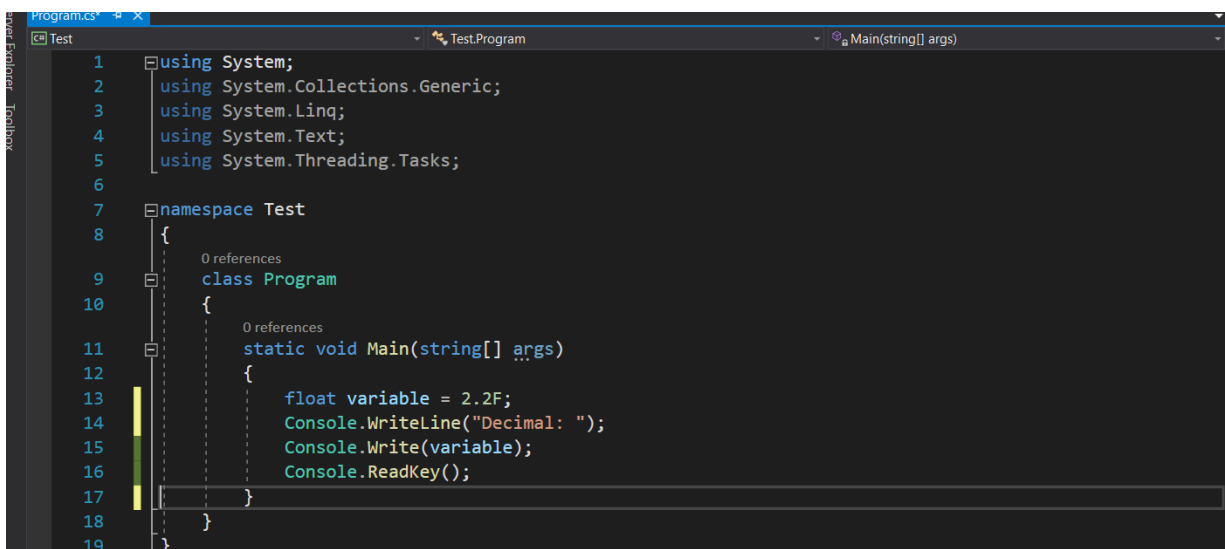
## Lab Manual 2

### Float Datatype

The Float datatype is slightly different in syntax as we need to define the variable explicitly by adding an “F” at the end of the given value.

Look at the following code snippet to understand the use of the “**Float**” datatype in C#.

#### Code:

A screenshot of the Visual Studio code editor. The file is named 'Program.cs' and is part of a project 'Test'. The code defines a namespace 'Test' containing a class 'Program'. Inside the 'Main' method, a float variable is declared and assigned the value 2.2F. The variable is then printed to the console using 'Console.WriteLine' and 'Console.Write', and the program waits for a key press with 'Console.ReadKey()'.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Test
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            float variable = 2.2F;
14            Console.WriteLine("Decimal: ");
15            Console.Write(variable);
16            Console.ReadKey();
17        }
18    }
19 }
```

#### Output:

A screenshot of a console window titled 'C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe'. The output shows the text 'Decimal:' followed by '2.2' on the next line.

```
Decimal:
2.2
```

# Object Oriented Programming

## Lab Manual 2

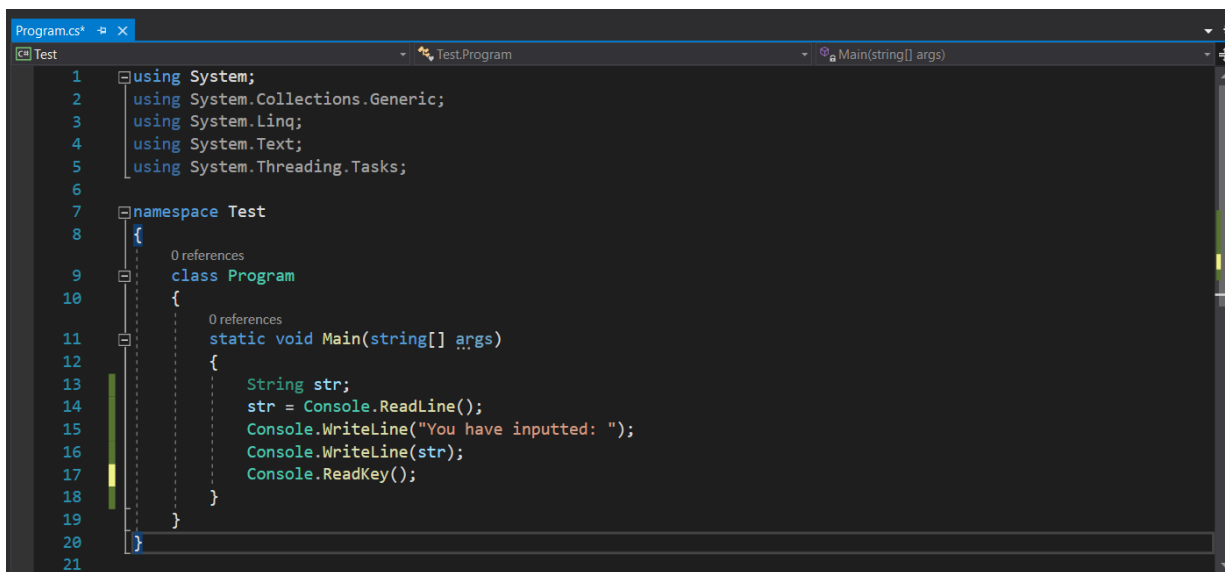
### Getting Input from User

The most important thing to remember is that every time you will take input from the user, it will be considered as String type input by the C#. This is the by-default behavior of the language.

“**Console.ReadLine()**” is used to take input from the user.

Look at the following code snippet to understand this functionality.

#### Code:



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Test
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             String str;
14             str = Console.ReadLine();
15             Console.WriteLine("You have inputted: ");
16             Console.WriteLine(str);
17             Console.ReadKey();
18         }
19     }
20 }
21
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
This is input from user
You have inputted:
This is input from user
_
```

# Object Oriented Programming

## Lab Manual 2

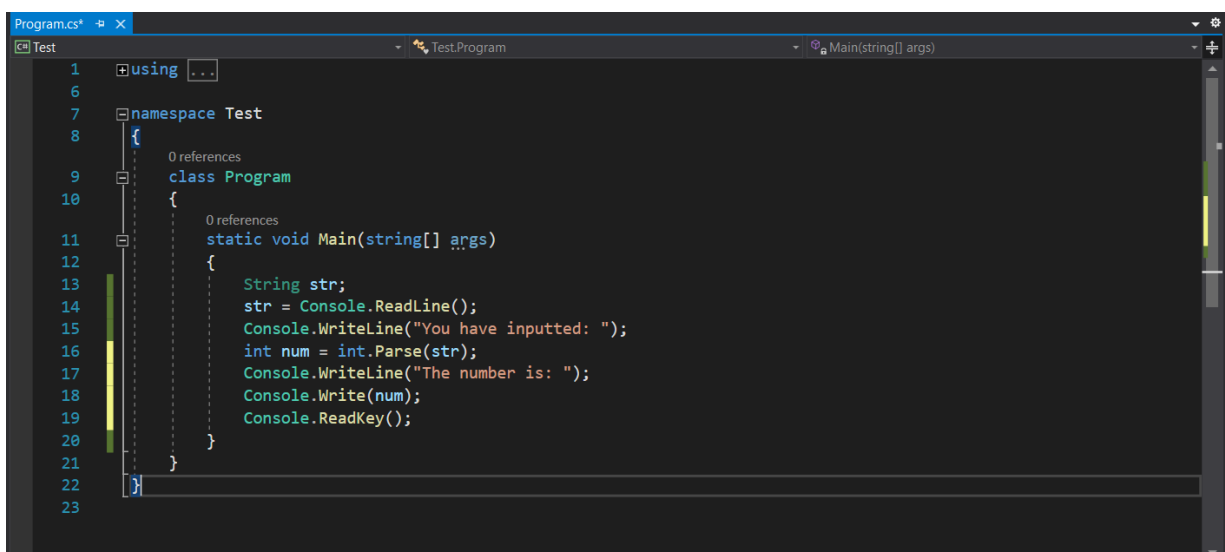
### Taking input as Integer from User

Now, we know that every time we will take input from the user, it will be taken as “string” datatype. However, what if we need to take the input as “int” or “float”.

To solve this problem, we will use the built-in functionality of typecasting in C#. It will convert the given string type input into the required format.

Look at the following code snippet to grasp the understanding of this concept.

#### Code:



```
1  using ...
6
7  namespace Test
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             String str;
16             str = Console.ReadLine();
17             Console.WriteLine("You have inputted: ");
18             int num = int.Parse(str);
19             Console.WriteLine("The number is: ");
20             Console.Write(num);
21             Console.ReadKey();
22         }
23     }
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
178
You have inputted:
The number is:
178
```



# Object Oriented Programming

## Lab Manual 2

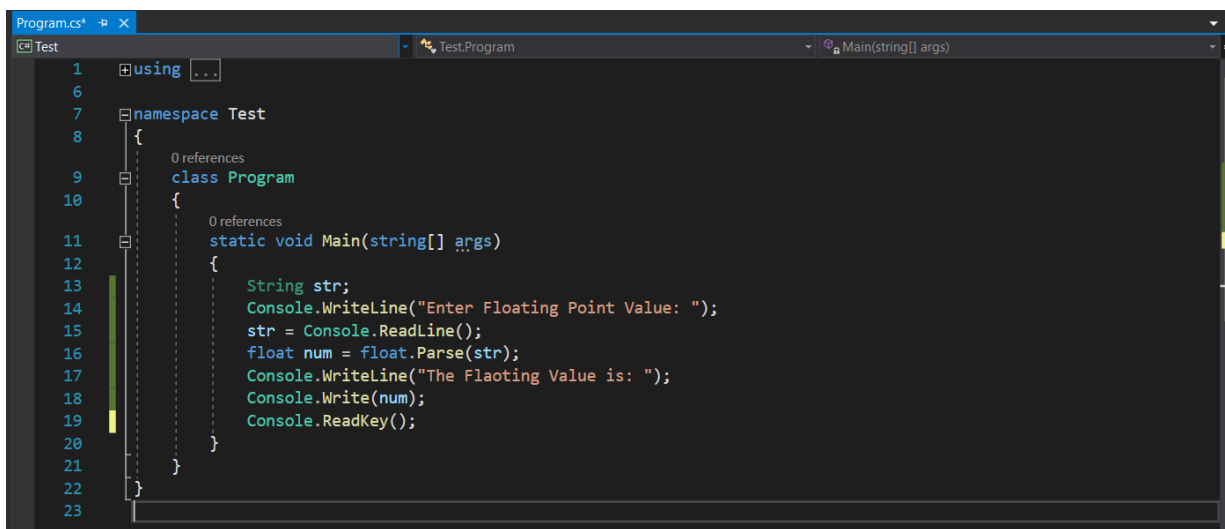
### Taking input as a float from User

Just like the previous phenomenon, we need to convert the received input from the user into Float as well if it is the kind of data that we want to use in our program.

We would use the “**float.Parse()**” function to convert the received input through the “**Console.ReadLine()**” function.

Look at the following code snippet to grasp the understanding of this concept.

#### Code:



```
1  using ...
6
7  namespace Test
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             String str;
14             Console.WriteLine("Enter Floating Point Value: ");
15             str = Console.ReadLine();
16             float num = float.Parse(str);
17             Console.WriteLine("The Floating Value is: ");
18             Console.Write(num);
19             Console.ReadKey();
20         }
21     }
22 }
23
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
Enter Floating Point Value:
87.1212
The Floating Value is:
87.1212
```

# Object Oriented Programming

## Lab Manual 2

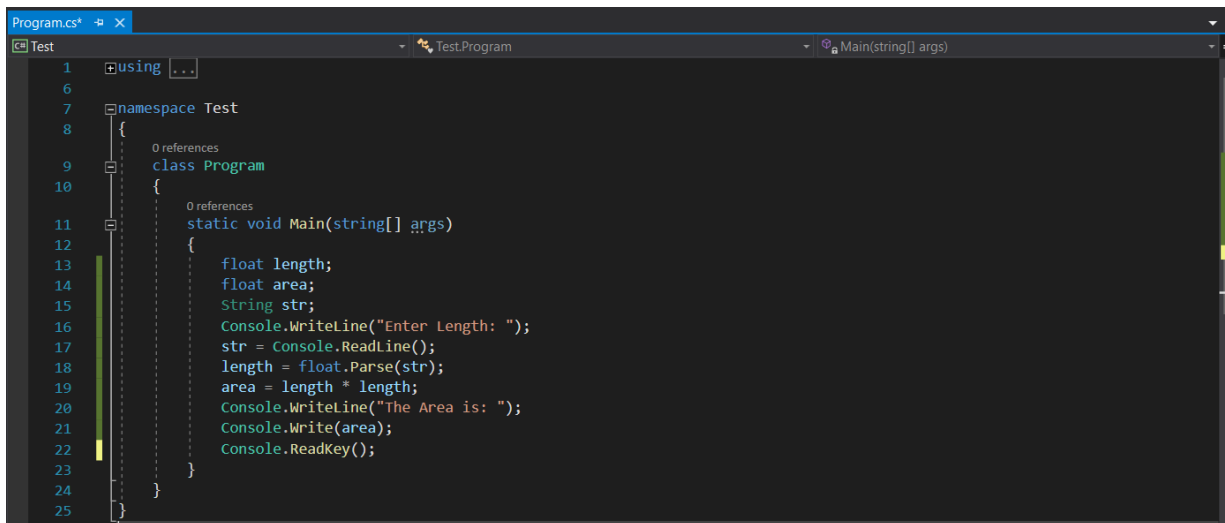
### Calculating the Area of Square

**Task:** Write a program that takes the length of one side of a square and prints the total area of the square on the screen.

**Solution:**

You can look below for a working example of this code.

#### Code:



```
1  using System;
2
3  namespace Test
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             float length;
12             float area;
13             String str;
14             Console.WriteLine("Enter Length: ");
15             str = Console.ReadLine();
16             length = float.Parse(str);
17             area = length * length;
18             Console.WriteLine("The Area is: ");
19             Console.Write(area);
20             Console.ReadKey();
21         }
22     }
23
24
25 }
```

#### Output:



```
C:\Users\HP\source\repos\Test\Test\bin\Debug\Test.exe
Enter Length:
4.5
The Area is:
20.25
```

Working Details for the above-mentioned program.

Actions		Description
	1	Creating a new project and putting the cursor inside the main function.
float area; float length; String str;	2	Declaring two variables for – <b>Length</b> and <b>Area</b> Declaring a String type variable for taking input.
Console.WriteLine("Enter Length: ");	3	Display on Screen to enter Length

# Object Oriented Programming

## Lab Manual 2

<code>str = Console.ReadLine();</code>	<b>4</b>	Take input from the user and storing into a variable
<code>length = float.Parse(str);</code>	<b>5</b>	Convert the input into Float type
<code>area = length * length;</code>	<b>6</b>	Calculation of Area
<code>Console.WriteLine("The Area is: ");</code> <code>Console.Write(area);</code>	<b>7</b>	Print the results on screen
<code>Console.ReadKey();</code>	<b>8</b>	Write the code for holding the output on screen until the user enters a key

**Good Luck and Best Wishes !!**

**Happy Coding ahead :)**