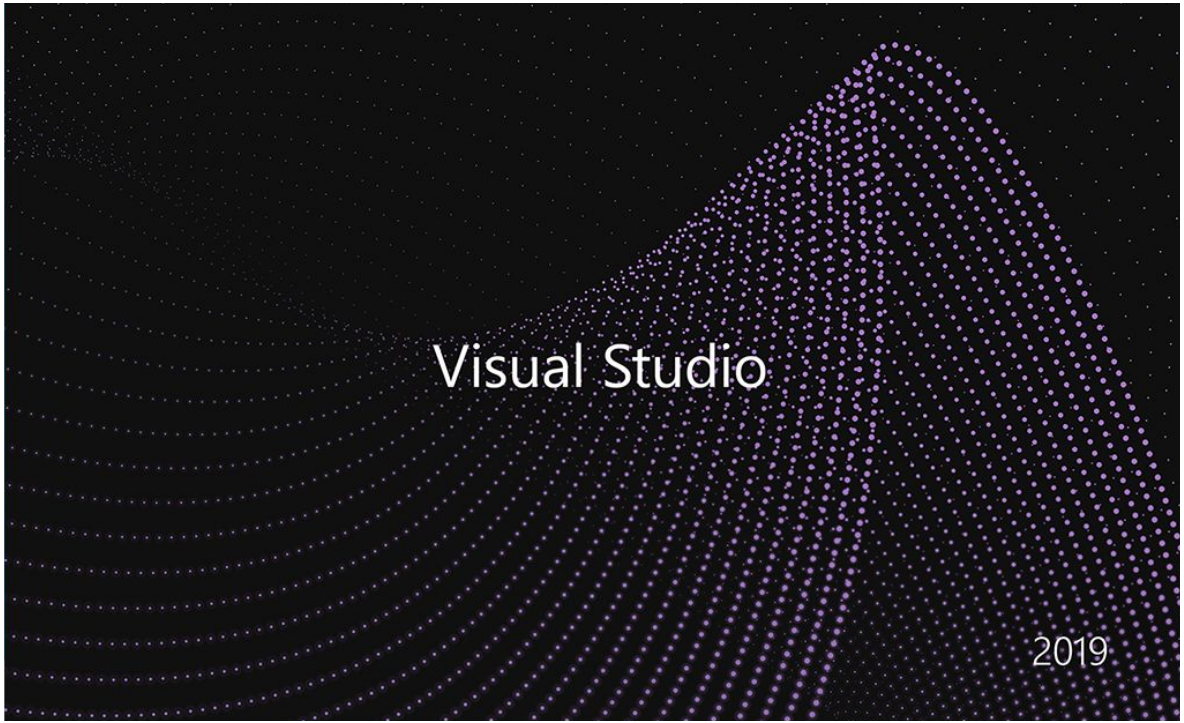
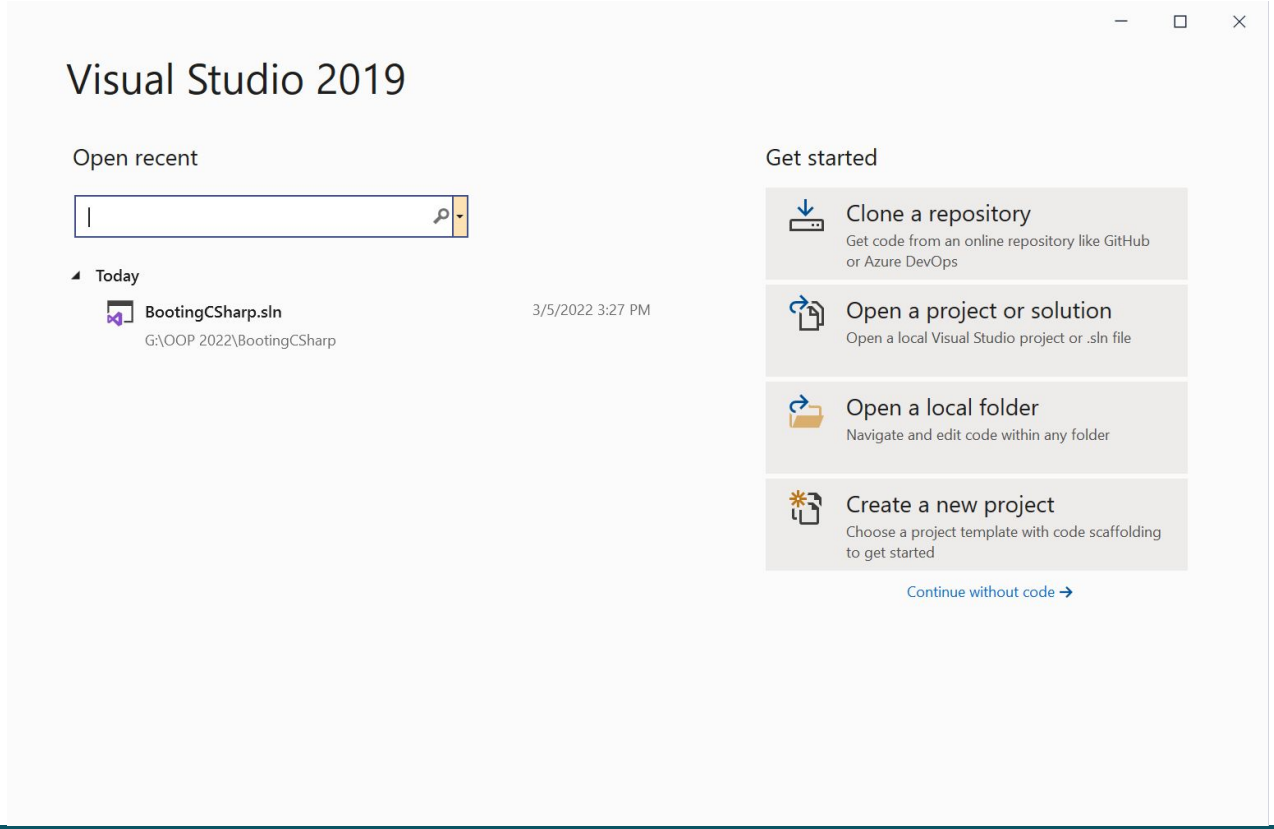
Booting on C#

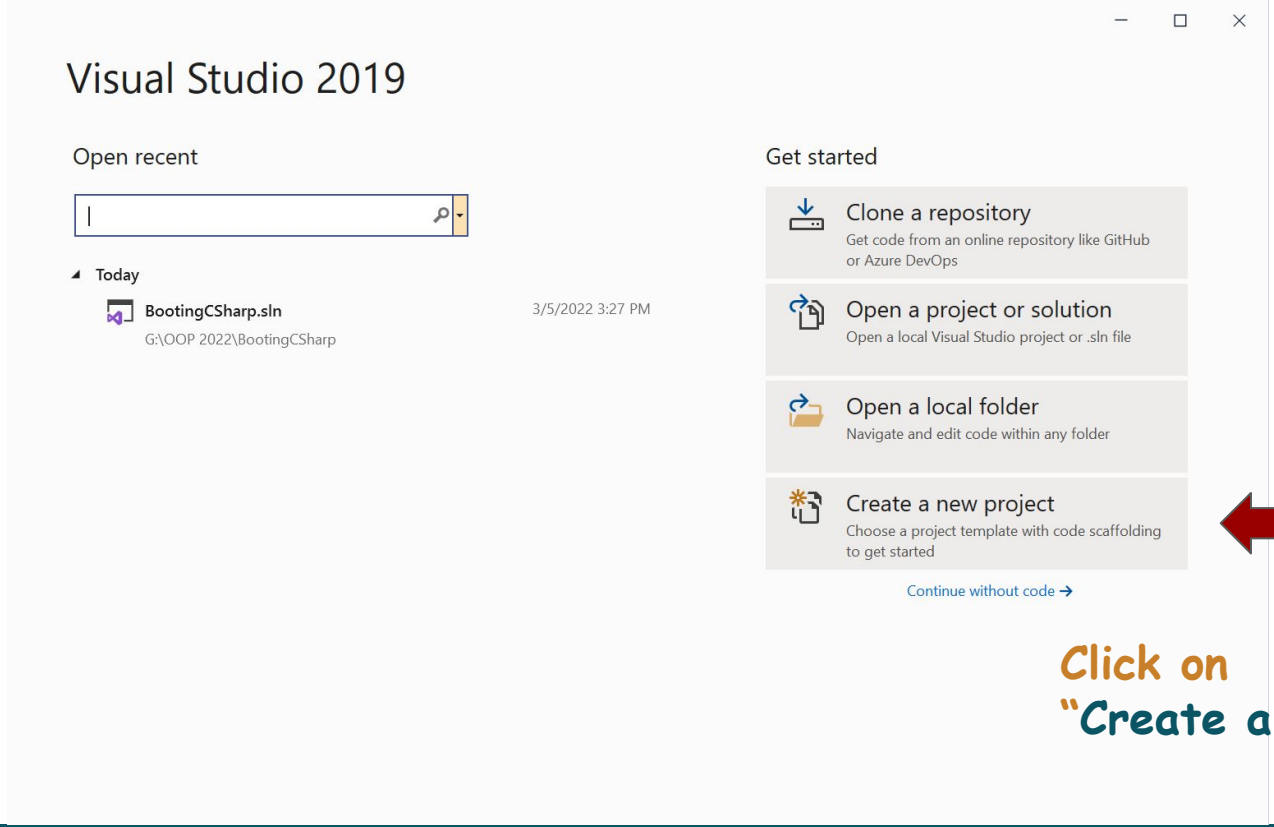
Visual Studio 2019 Community Edition



Visual Studio 2019 Community Edition

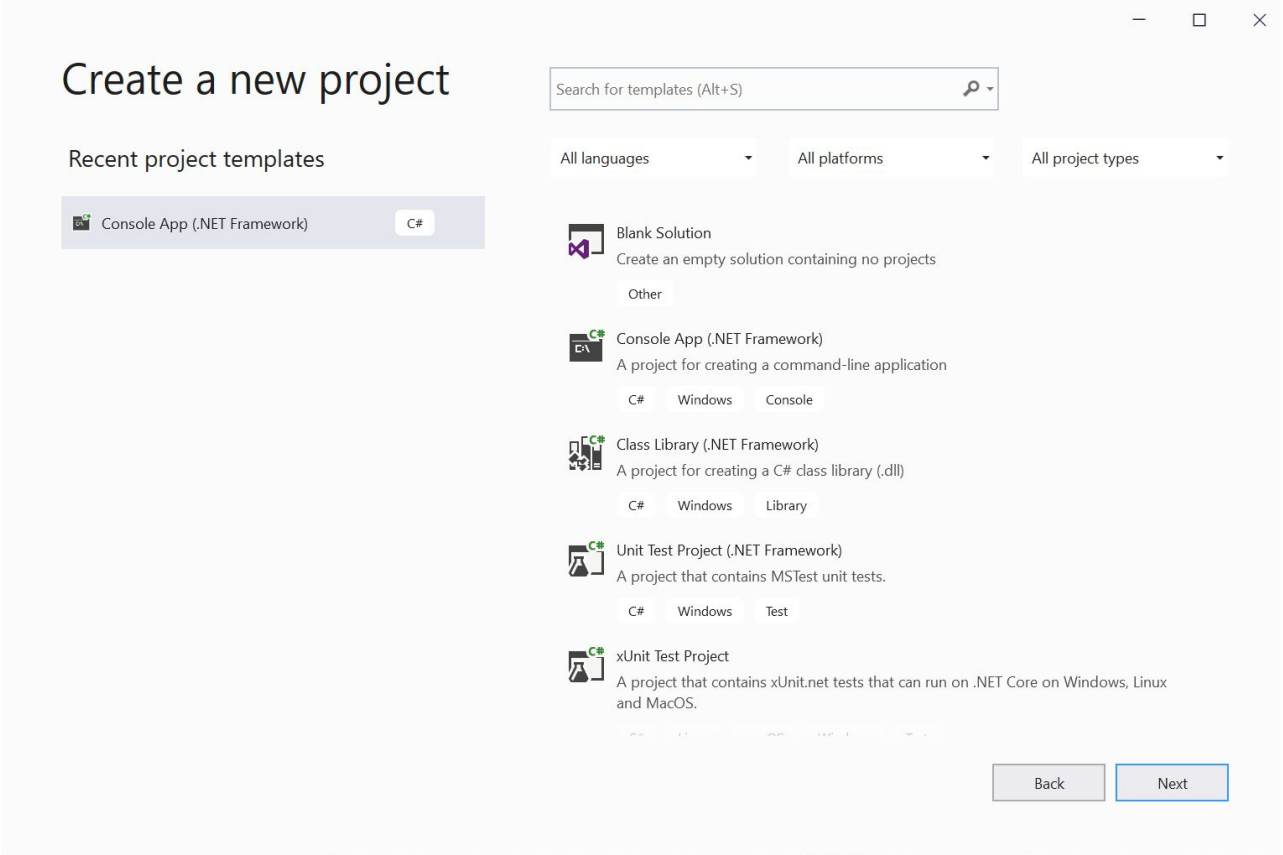


Visual Studio 2019 Community Edition

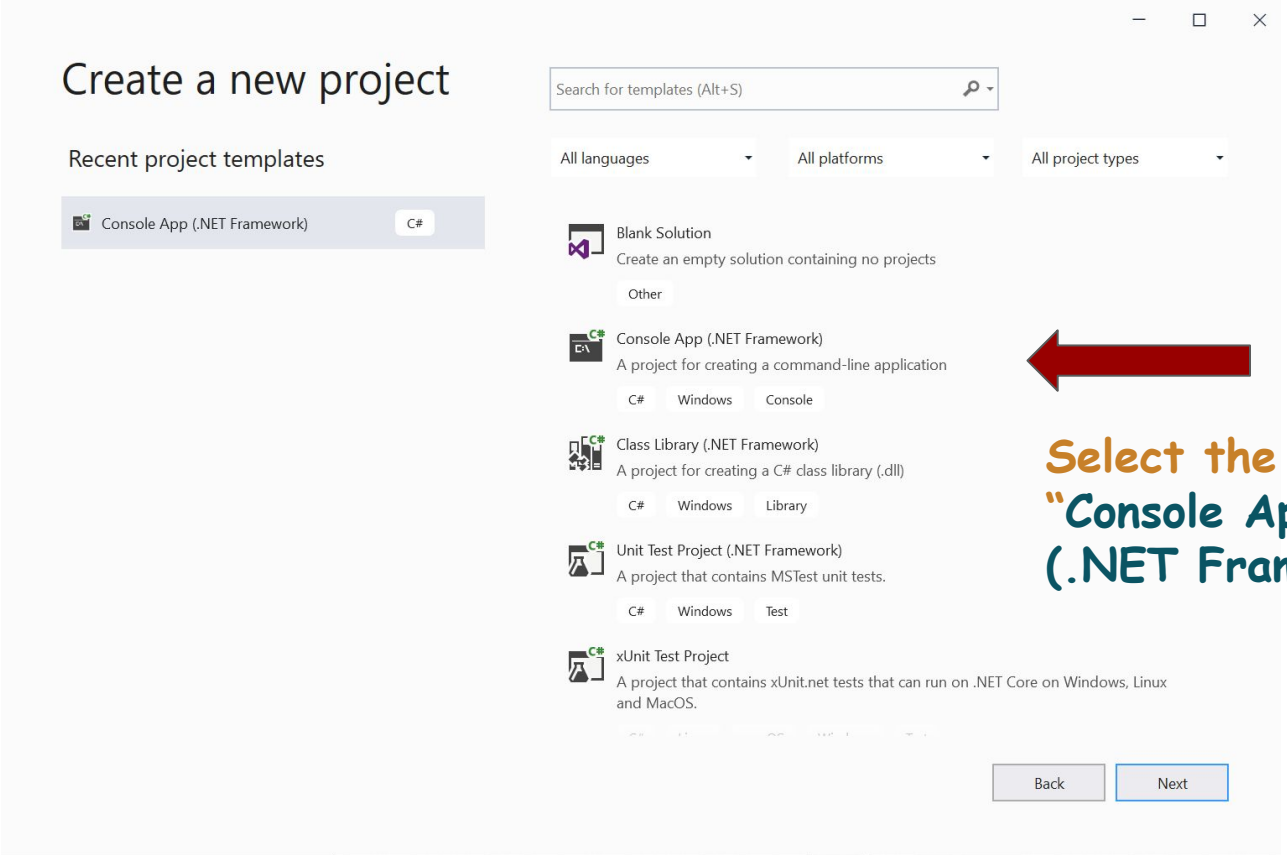


Click on "Create a new project"

Select the Project

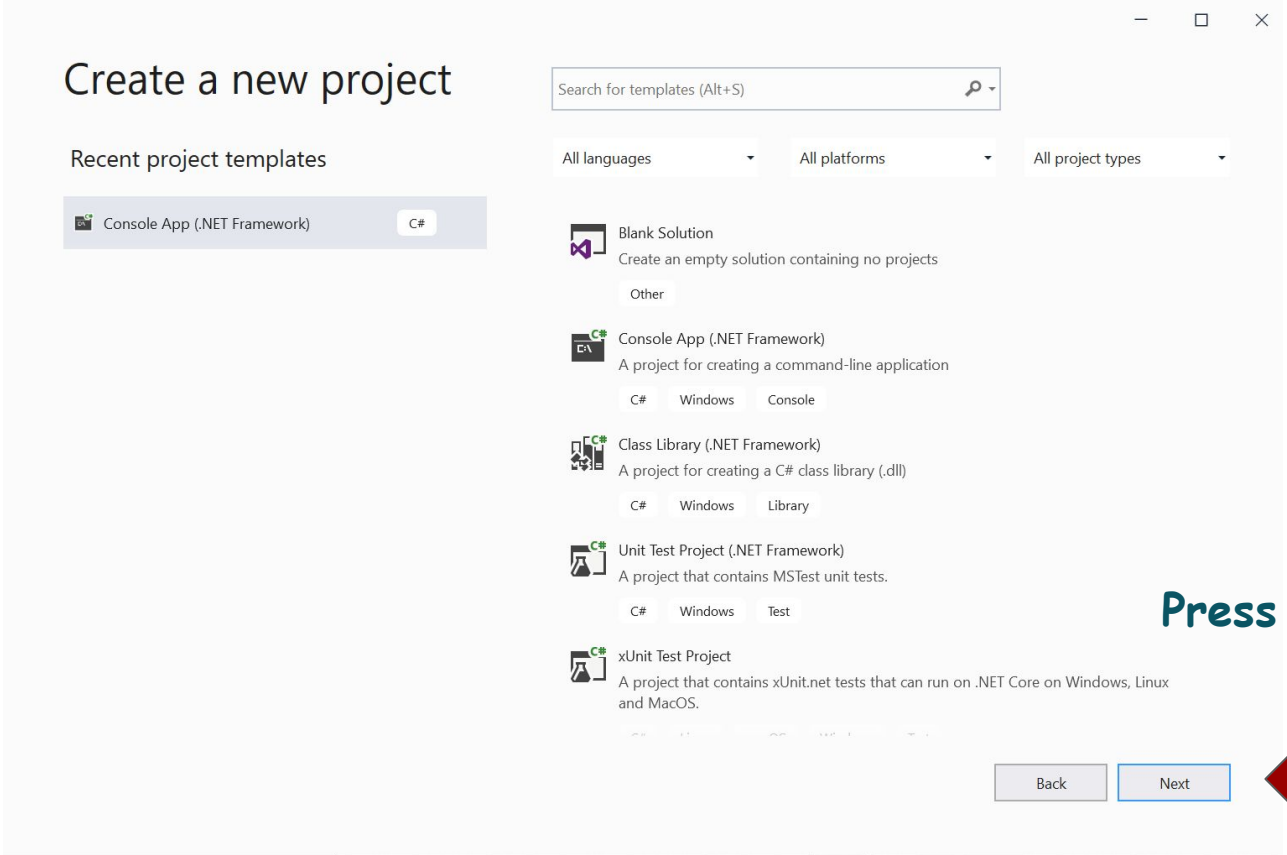


Select the Project



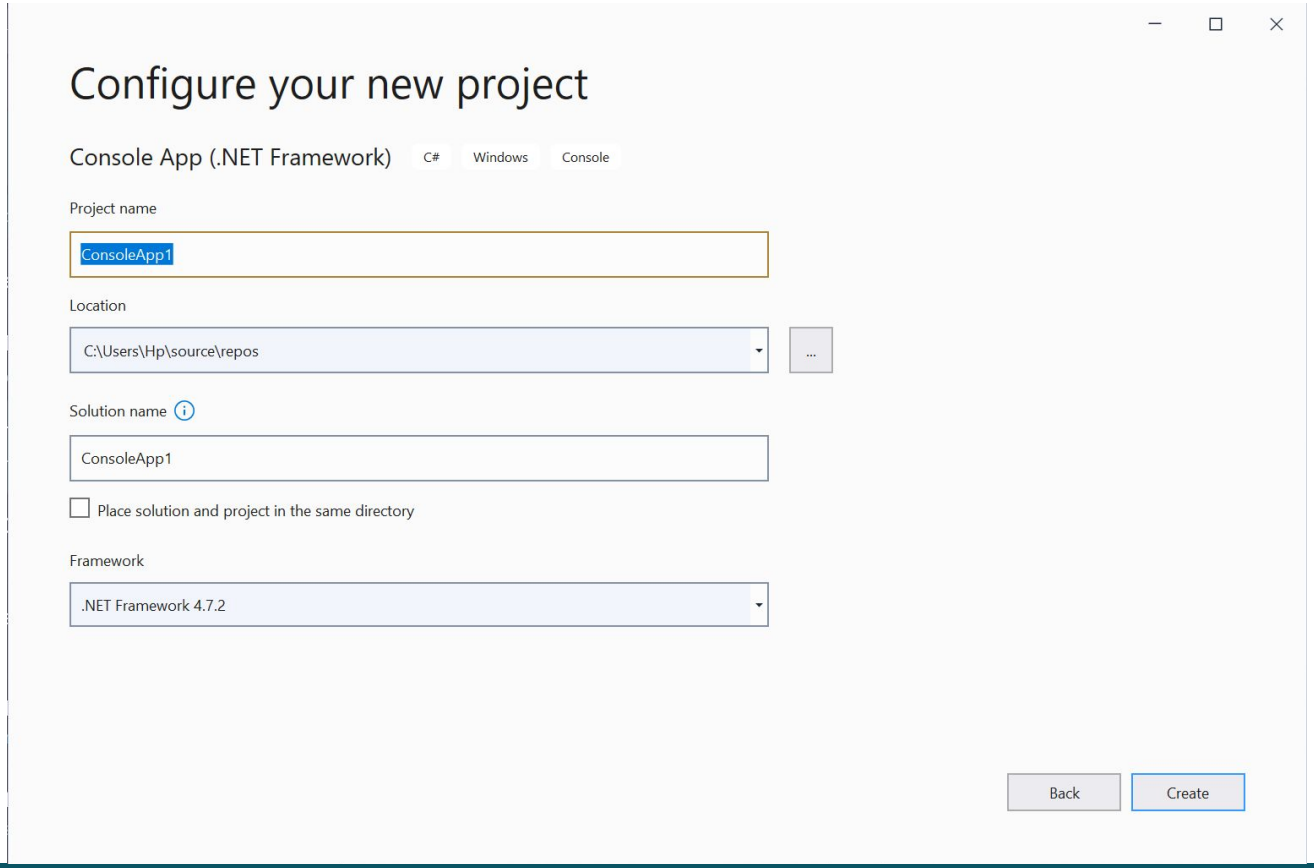
Select the Project "Console App (.NET Framework)"

Select the Project

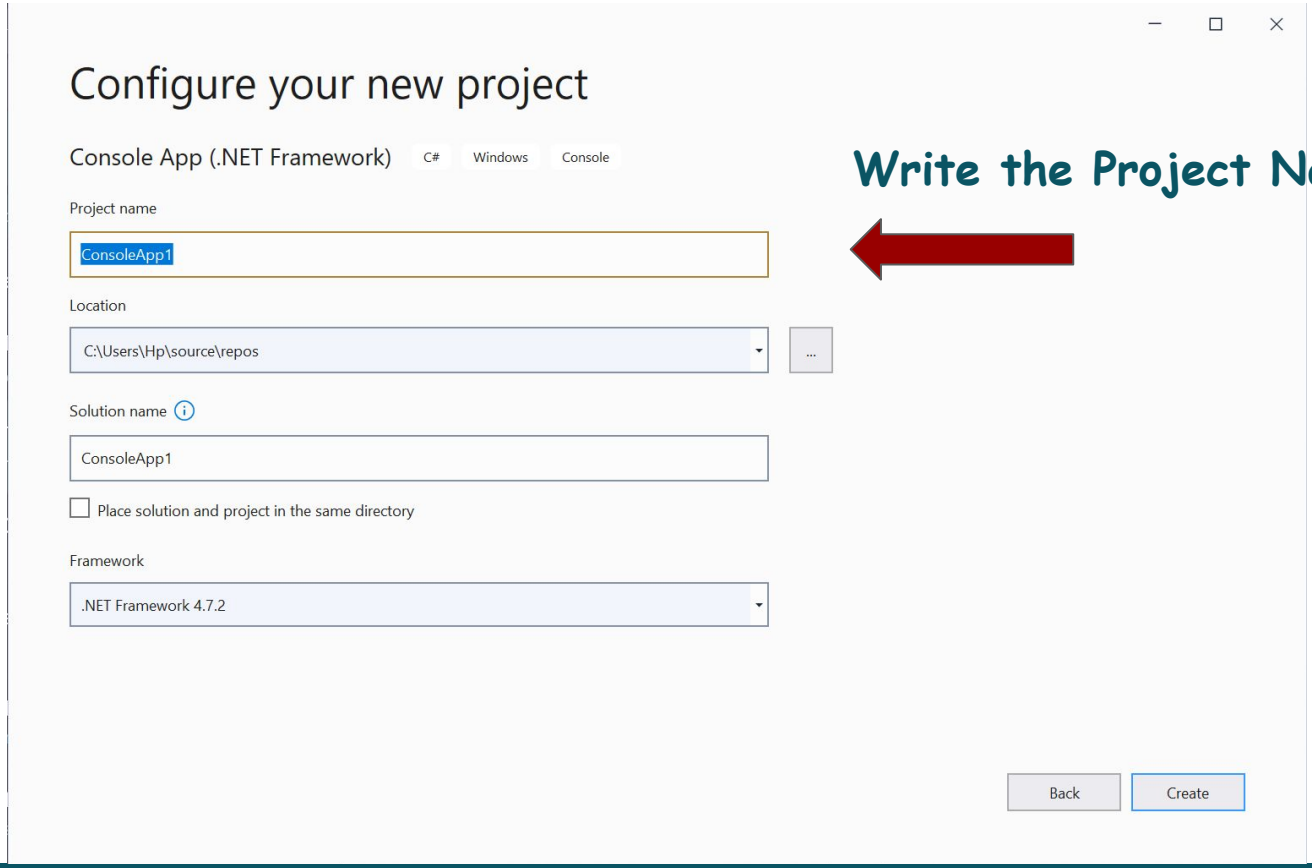


Press NEXT button

Configure the Project

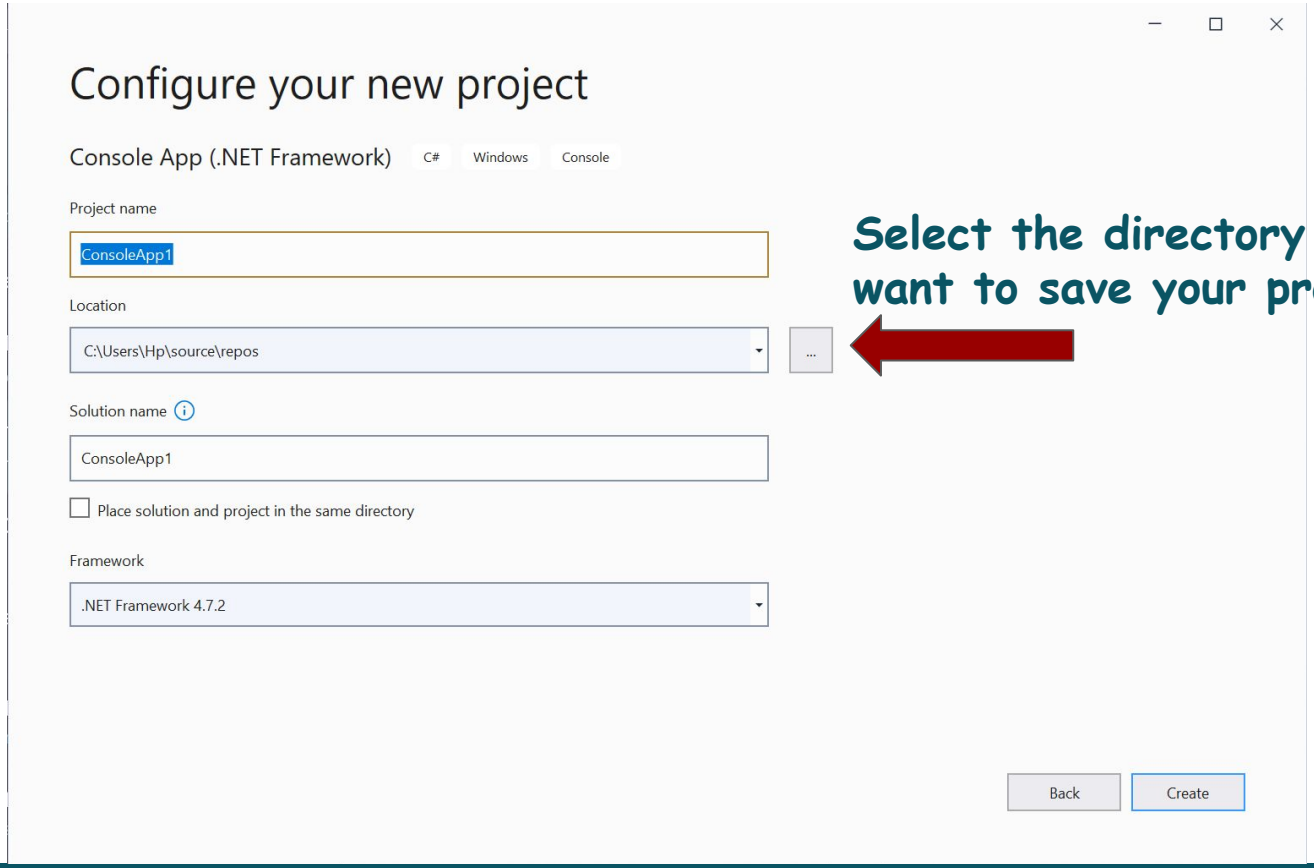


Configure the Project



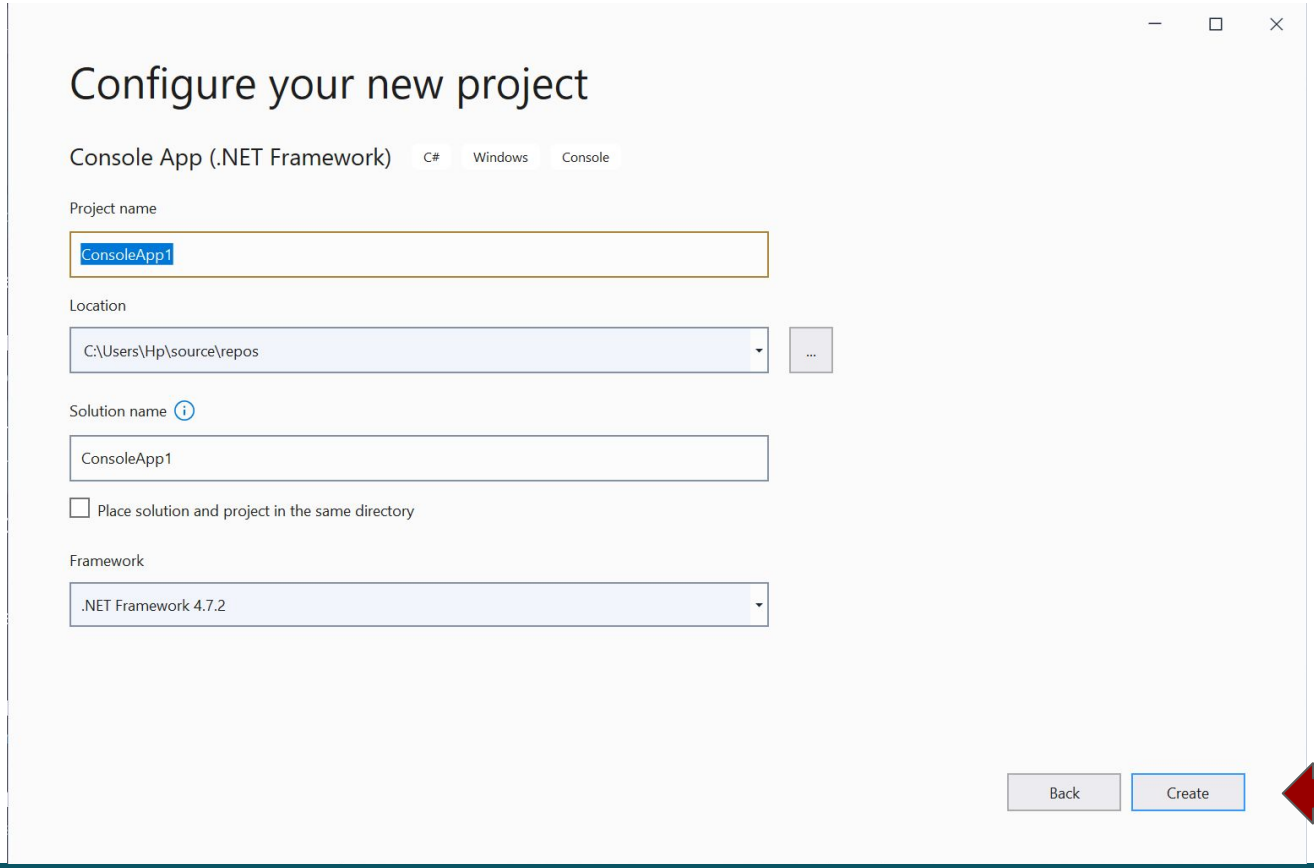
Write the Project Name

Configure the Project



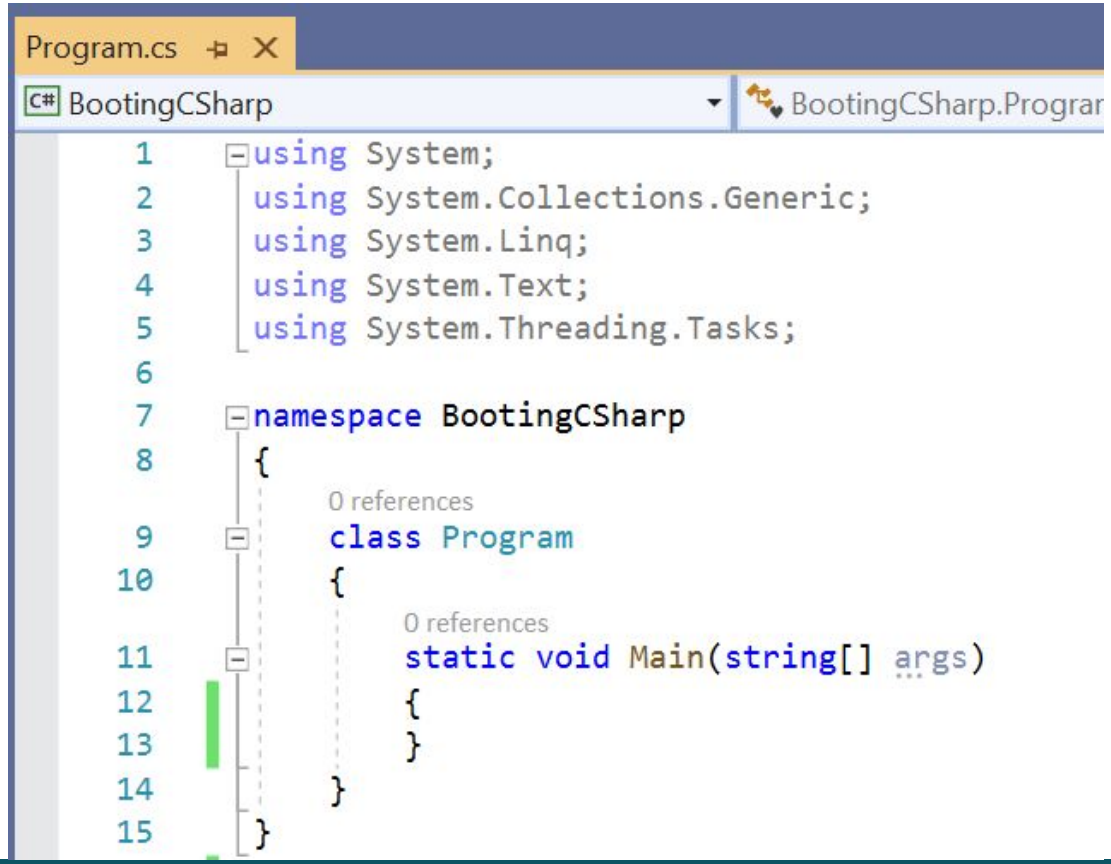
Select the directory where you want to save your project

Configure the Project

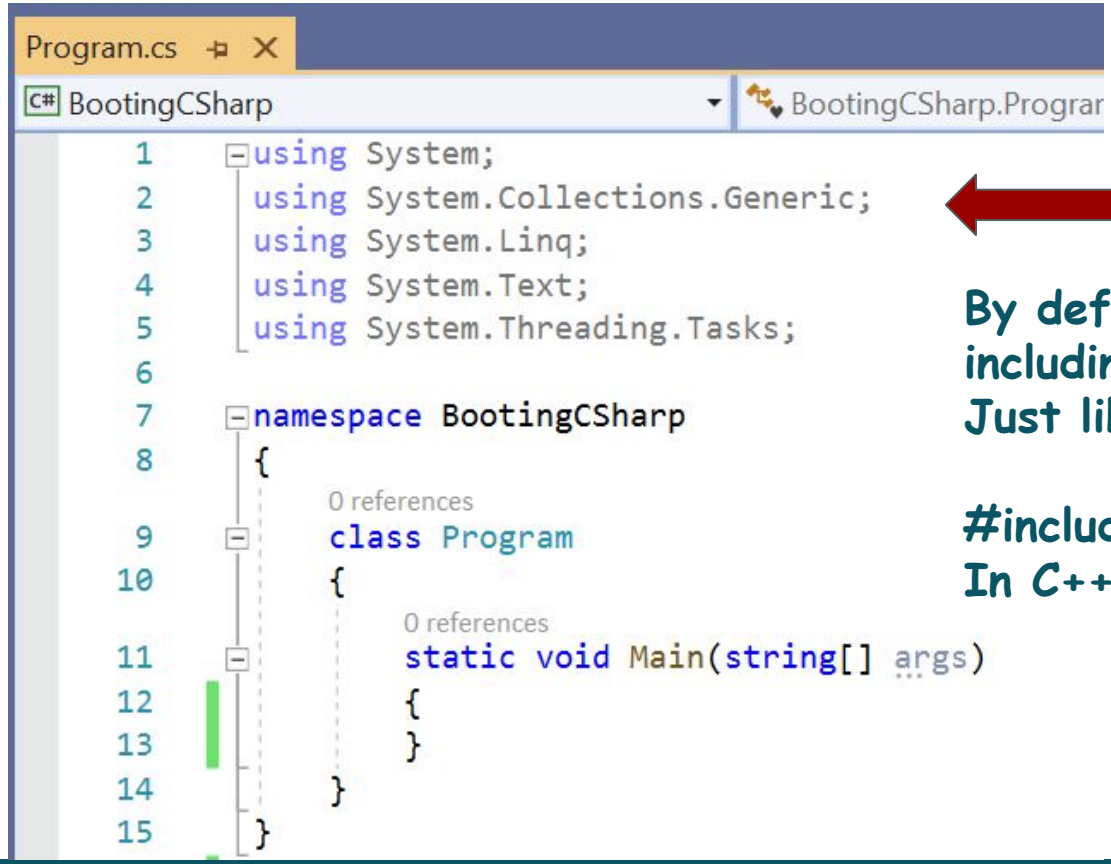


Press Create

C# Program Basic Structure



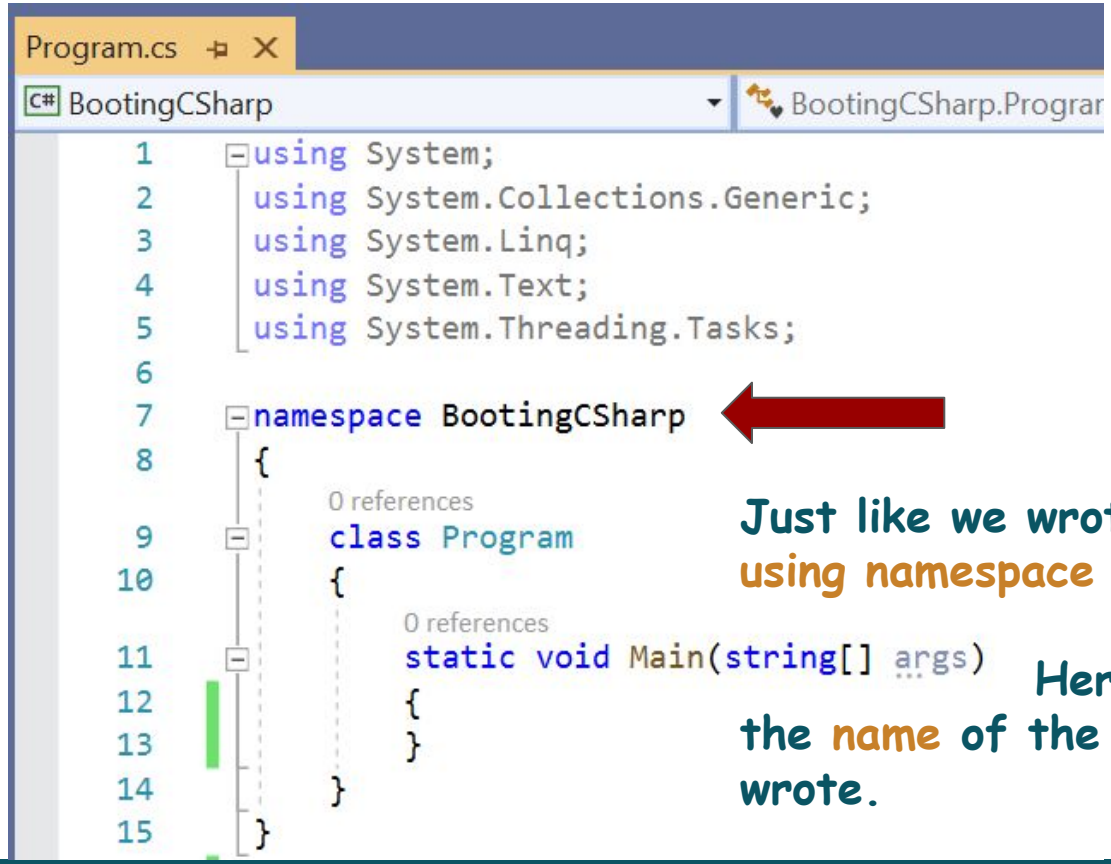
C# Program Basic Structure



By default it is including some libraries. Just like we did

#include<iostream>
In C++

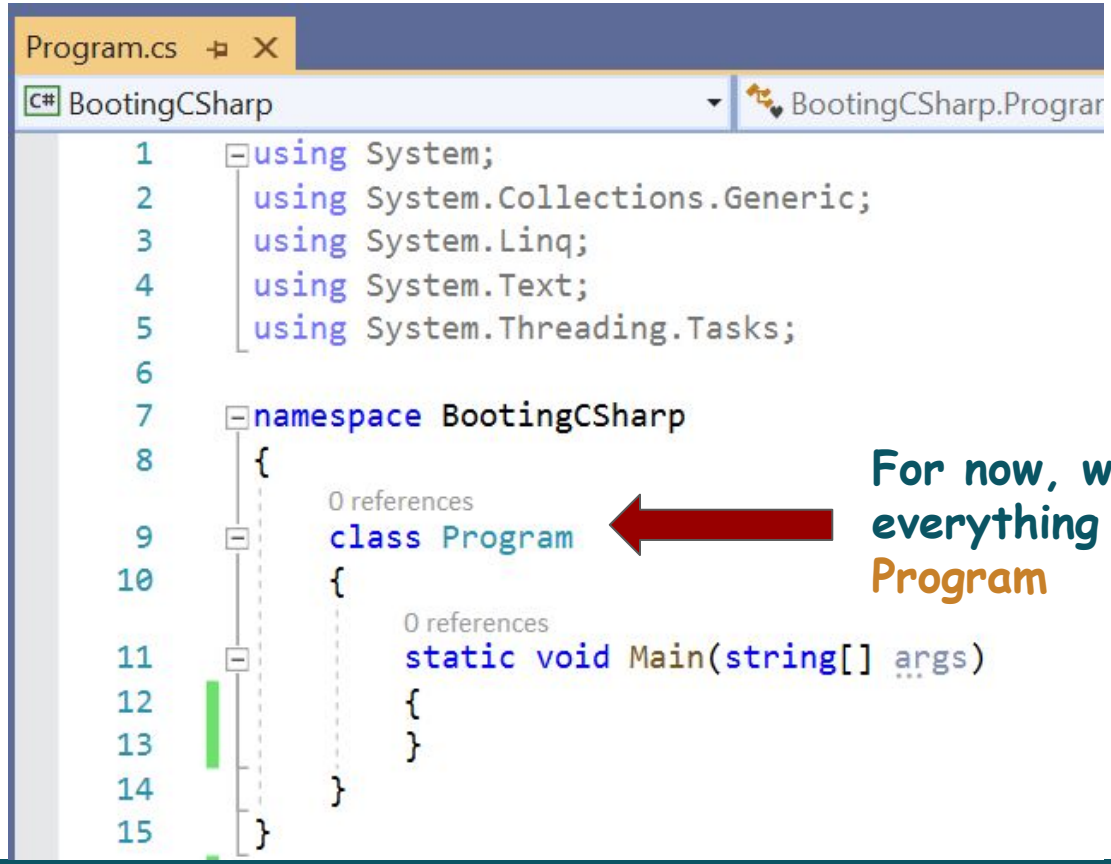
C# Program Basic Structure



Just like we wrote
using namespace std;

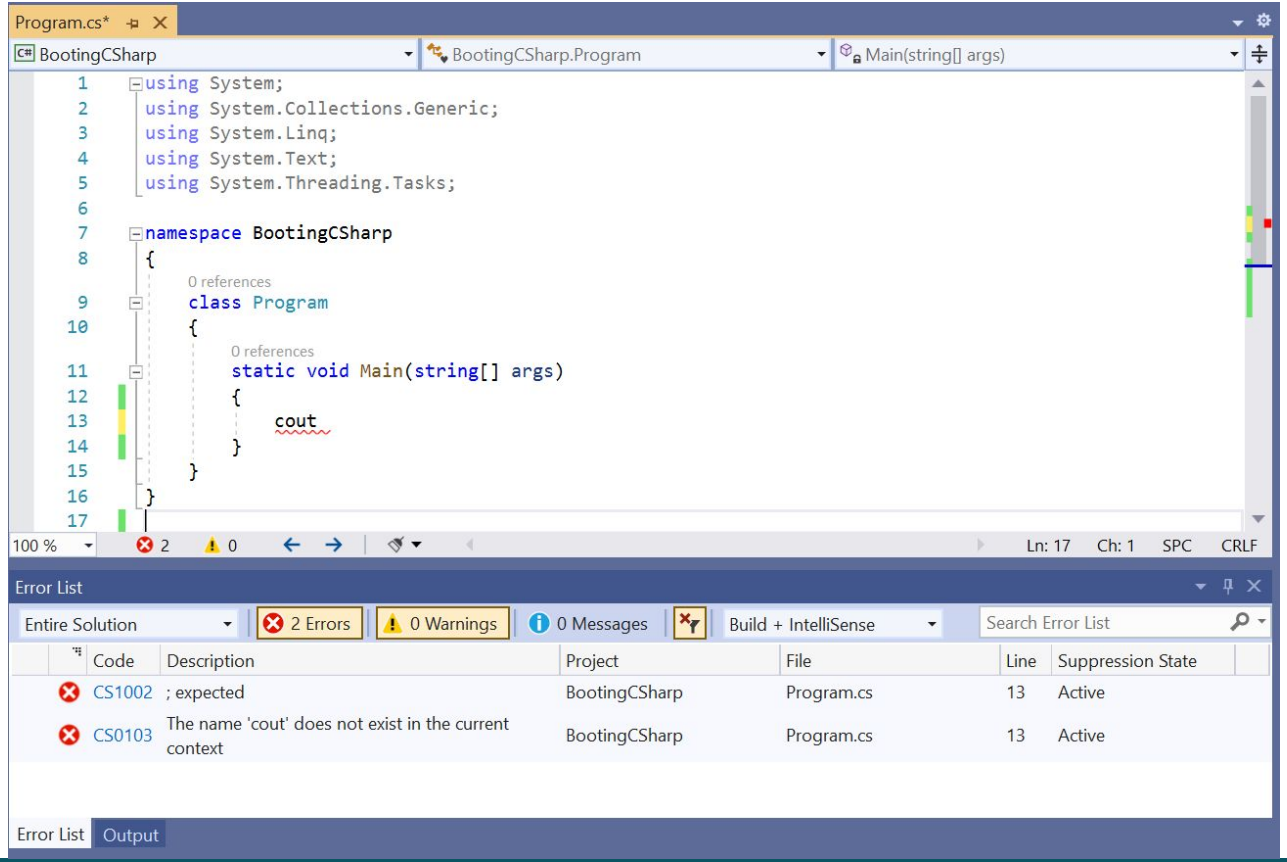
Here namespace is the name of the project that we wrote.

C# Program Basic Structure

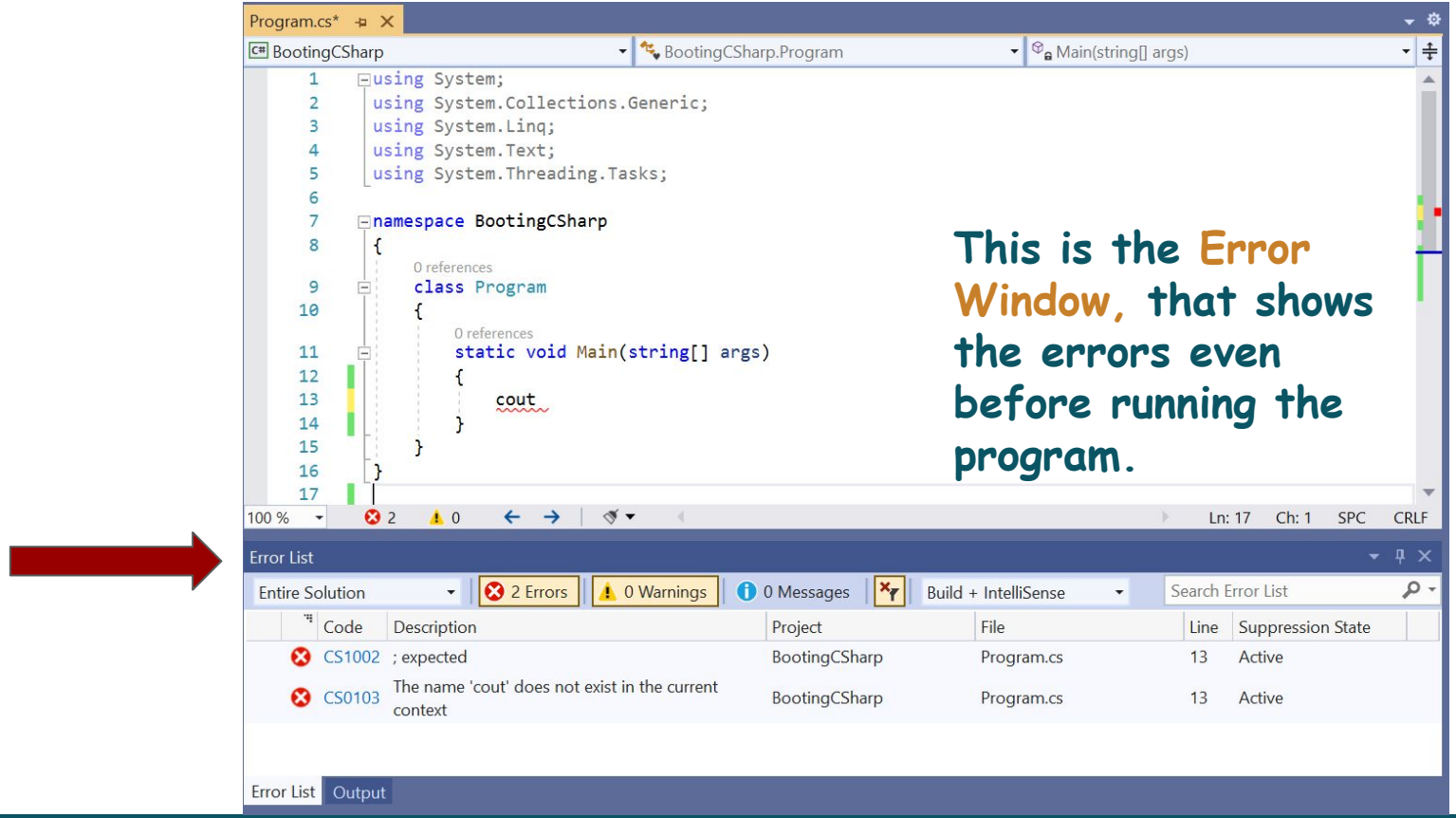


For now, we will write everything inside the class **Program**

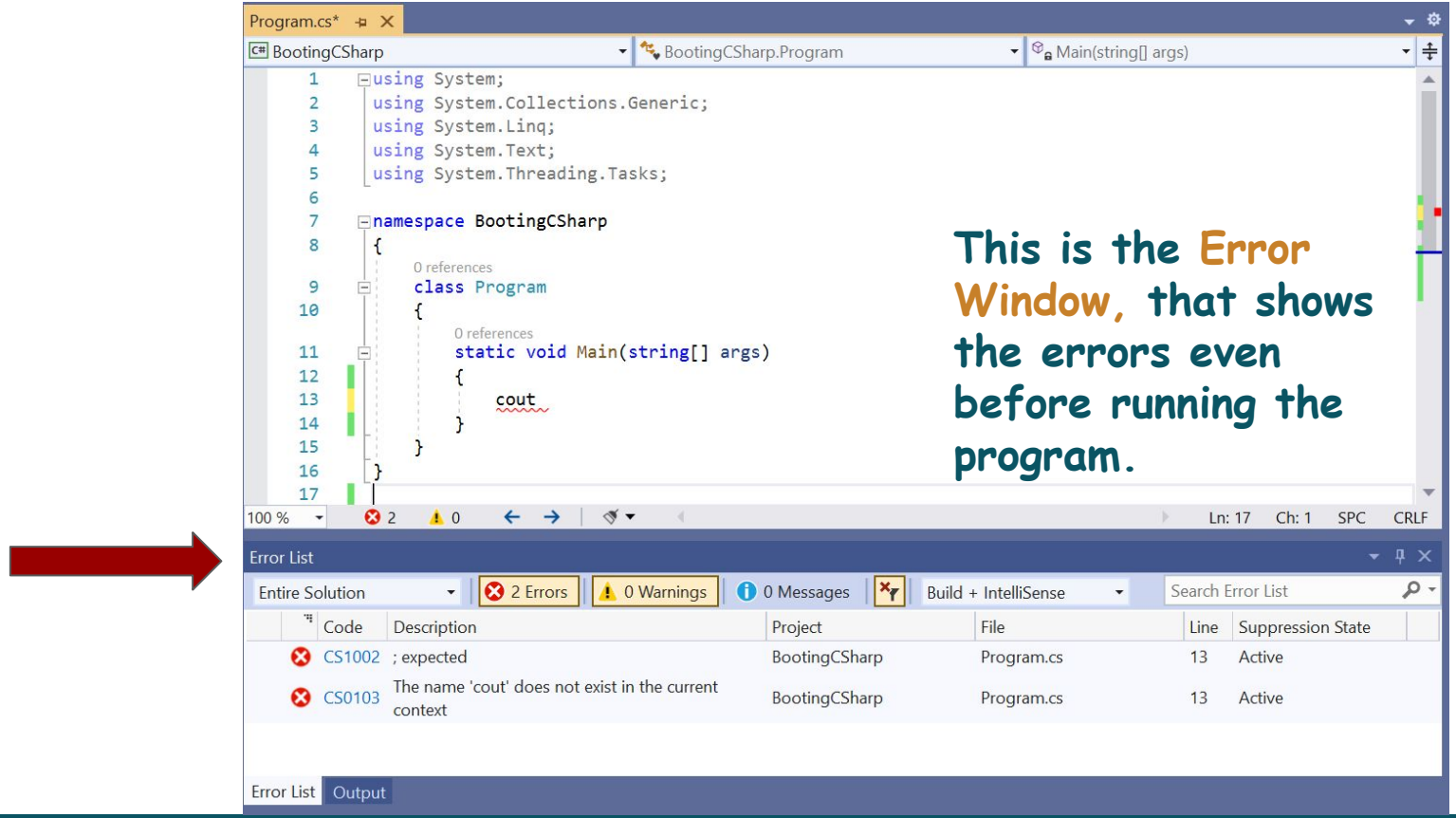
Visual Studio: Error Window



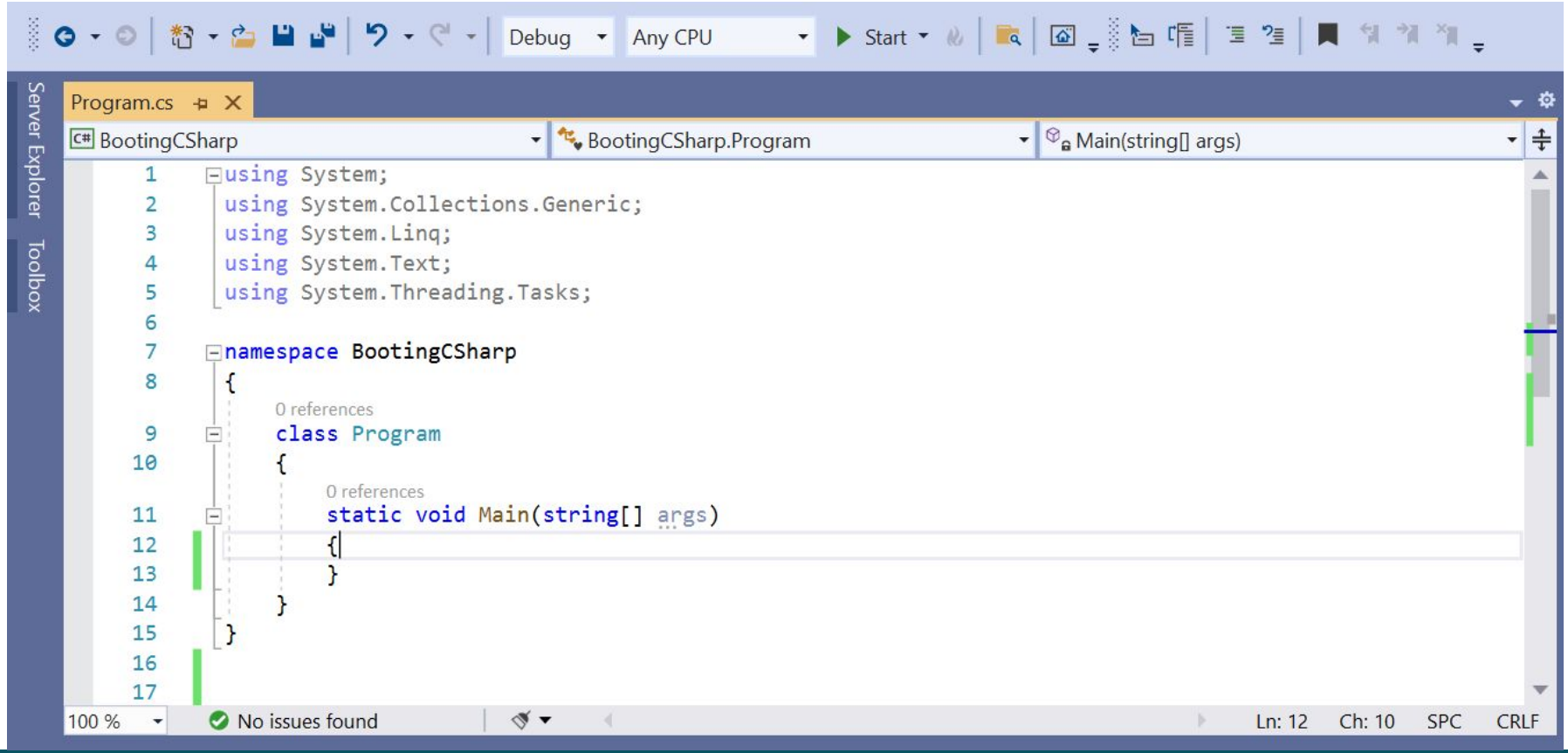
Visual Studio: Error Window



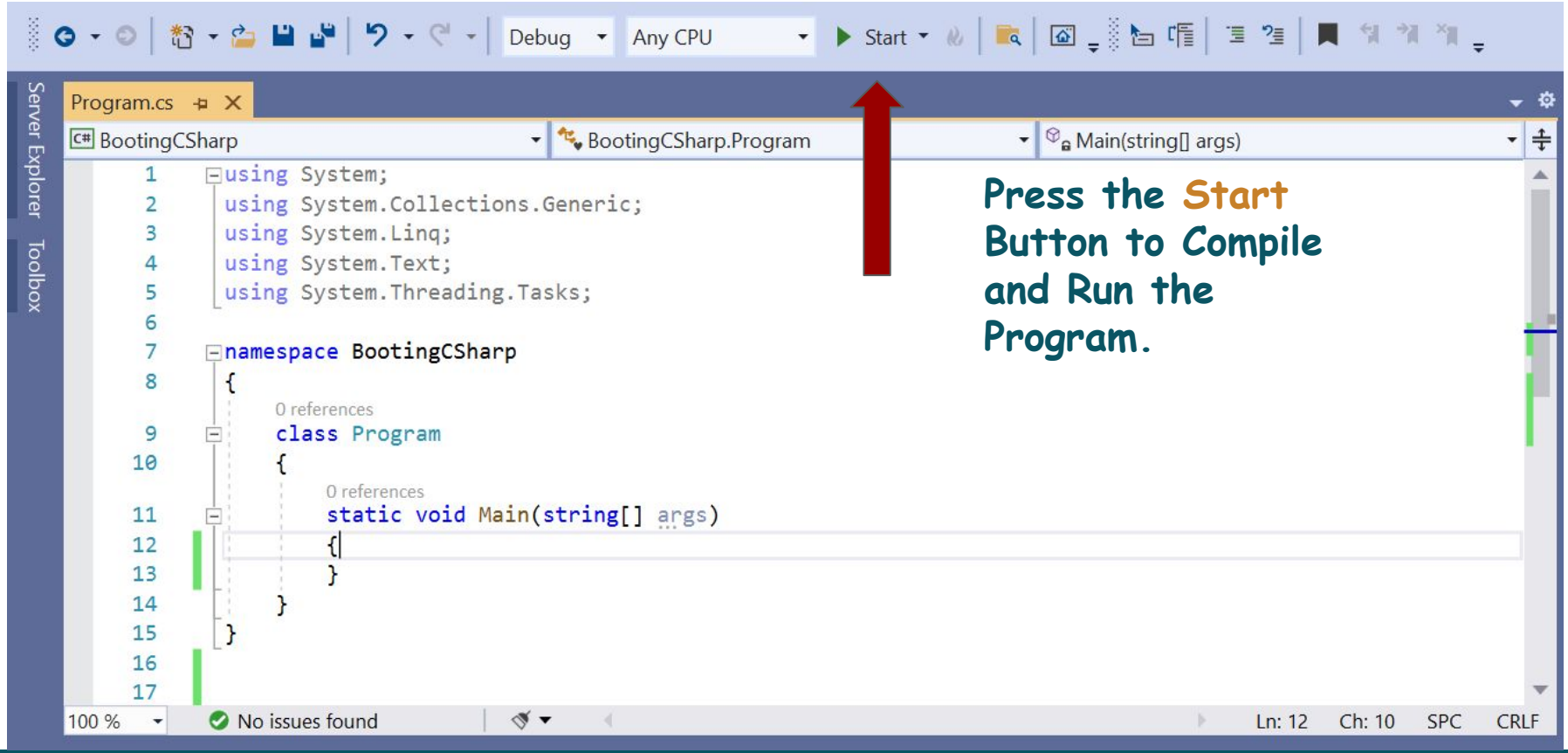
Visual Studio: Error Window



Visual Studio: Compile and Run

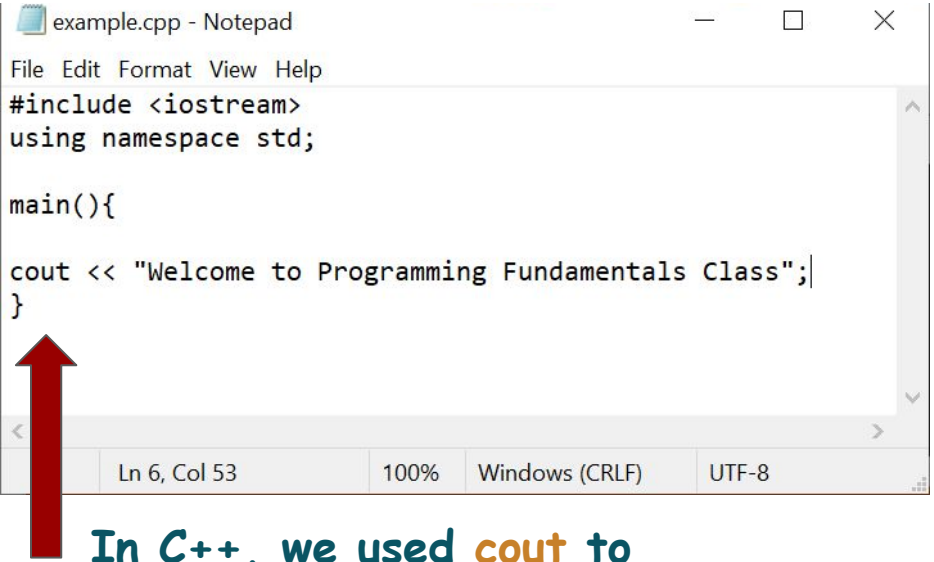


Visual Studio: Compile and Run



Displaying Output in C#

Displaying Output on Console



In C++, we used cout to display the output on the console.

Displaying Output on Console

example.cpp - Notepad

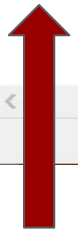
File Edit Format View Help

```
#include <iostream>
using namespace std;

main(){

cout << "Welcome to Programming Fundamentals Class";
}
```

Ln 6, Col 53 | 100% | Windows (CRLF) | UTF-8



In C++, we used cout to display the output on the console.

BootingCSharp | BootingCSharp.Program

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace BootingCSharp
8   {
        0 references
9       class Program
10      {
            0 references
11          static void Main(string[] args)
12          {
13              Console.Write("Welcome to OOP Class");
14          }
15      }
16  }
```

Displaying Output on Console

example.cpp - Notepad

File Edit Format View Help

```
#include <iostream>
using namespace std;

main(){

cout << "Welcome to Programming Fundamentals Class";
}
```

Ln 6, Col 53 | 100% | Windows (CRLF) | UTF-8

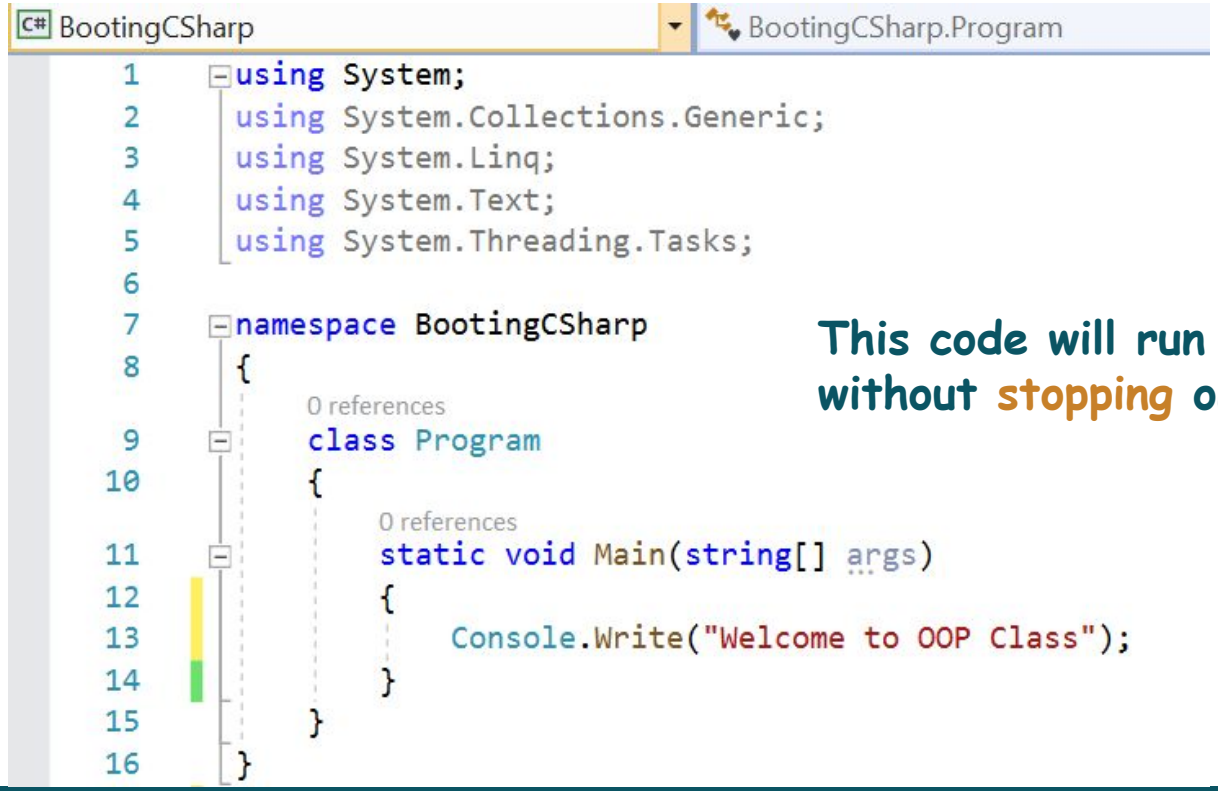
In C++, we used cout to display the output on the console.

BootingCSharp | BootingCSharp.Program

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace BootingCSharp
8   {
        0 references
9       class Program
10      {
            0 references
11          static void Main(string[] args)
12          {
13              Console.Write("Welcome to OOP Class");
14          }
15      }
16  }
```

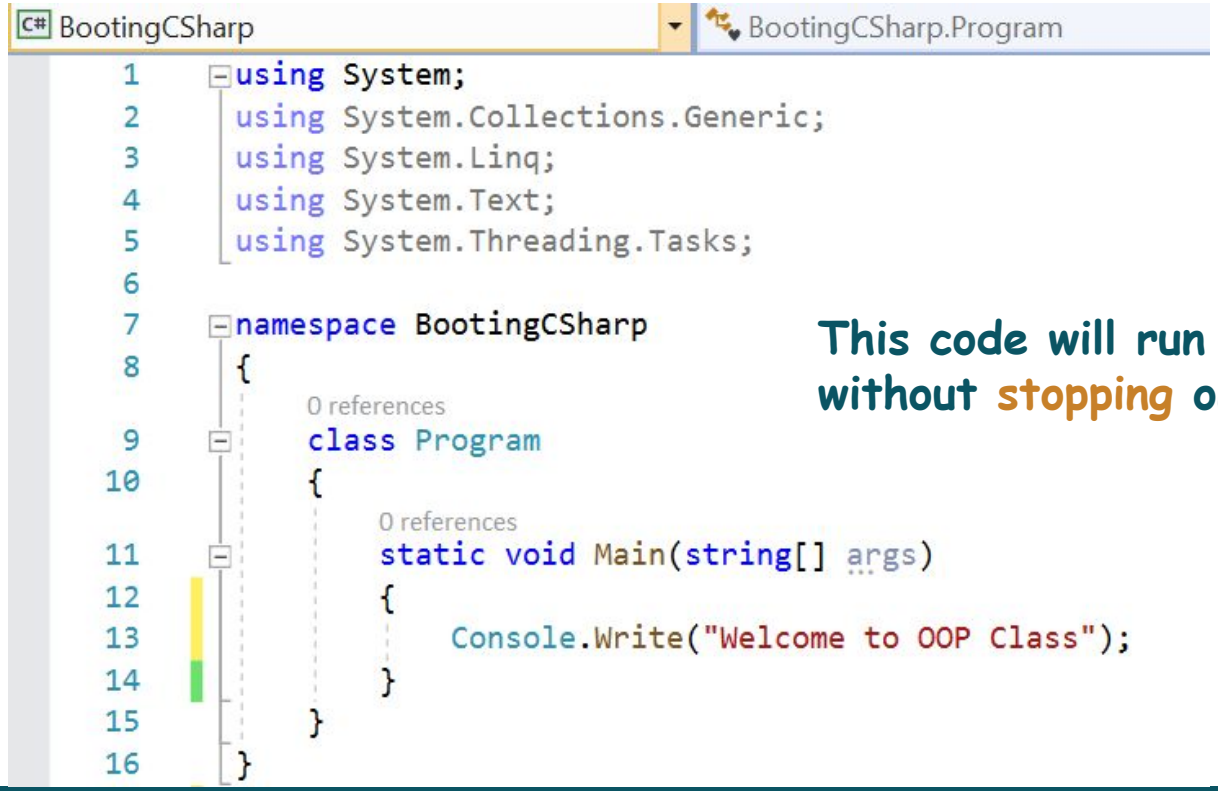
In C#, we use Console.Write(" ") to display the output on the console.

Displaying Output on Console



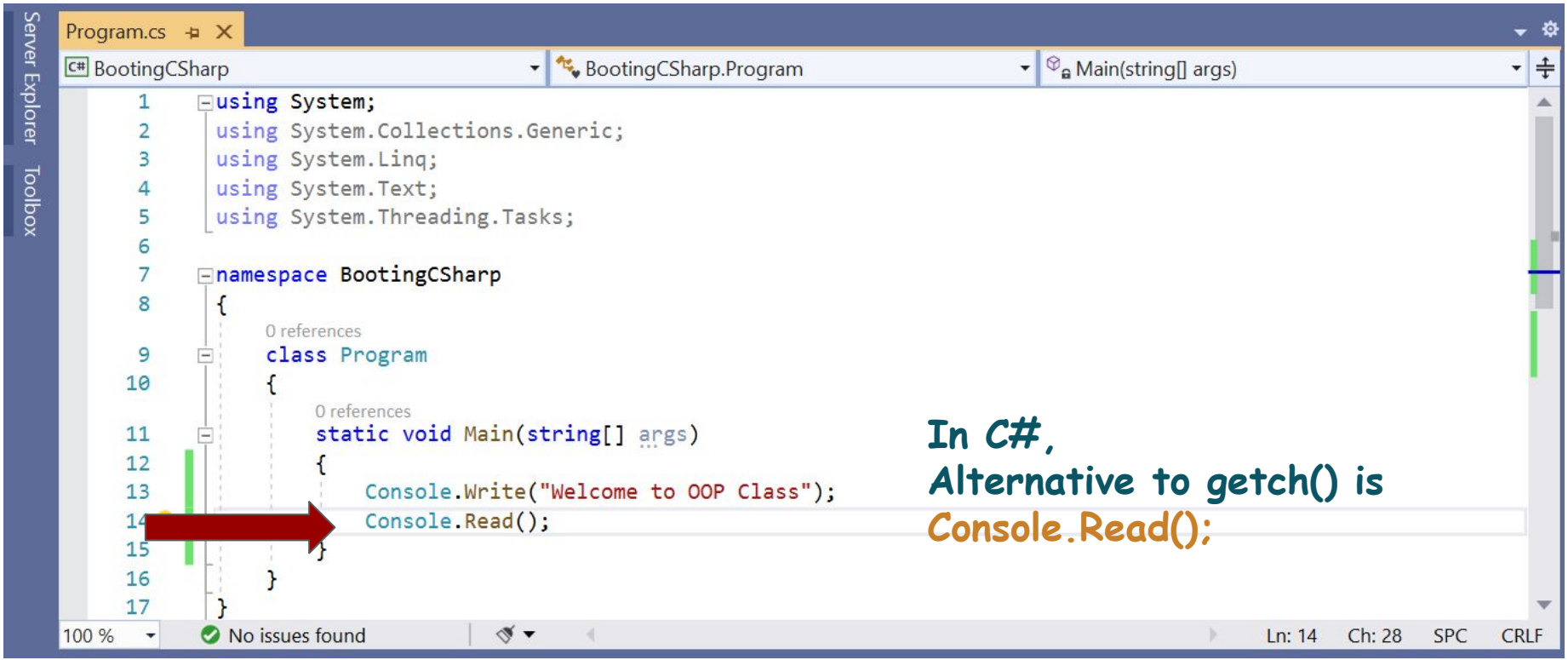
This code will run and terminate without stopping on the console.

Displaying Output on Console



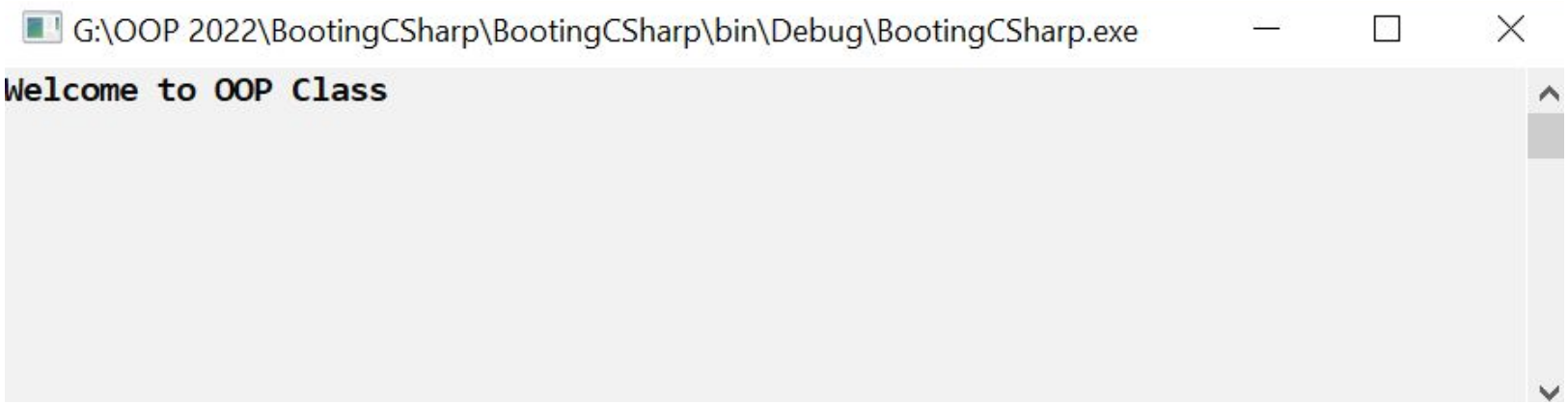
This code will run and terminate without stopping on the console.

Stopping at the Console

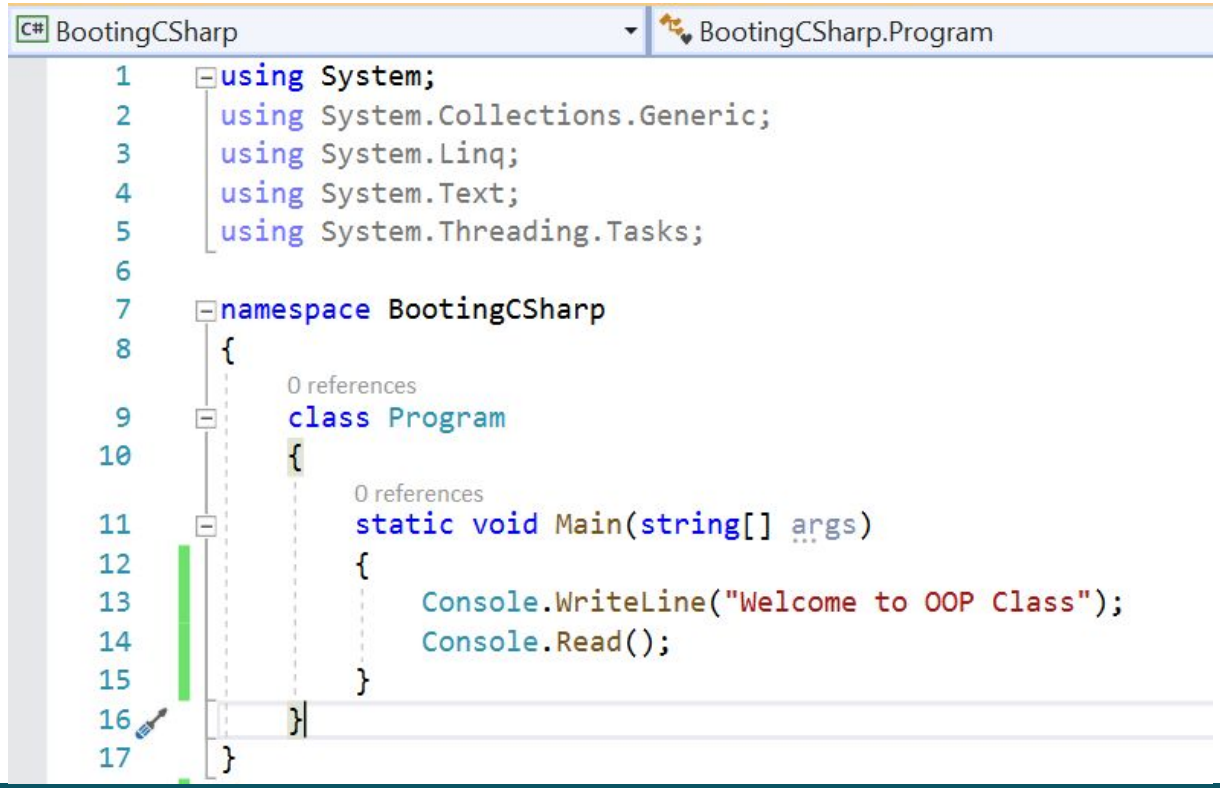


In C#,
Alternative to getch() is
Console.Read();

Output on the Console



Displaying Output on Console with endl



Displaying Output on Console with endl

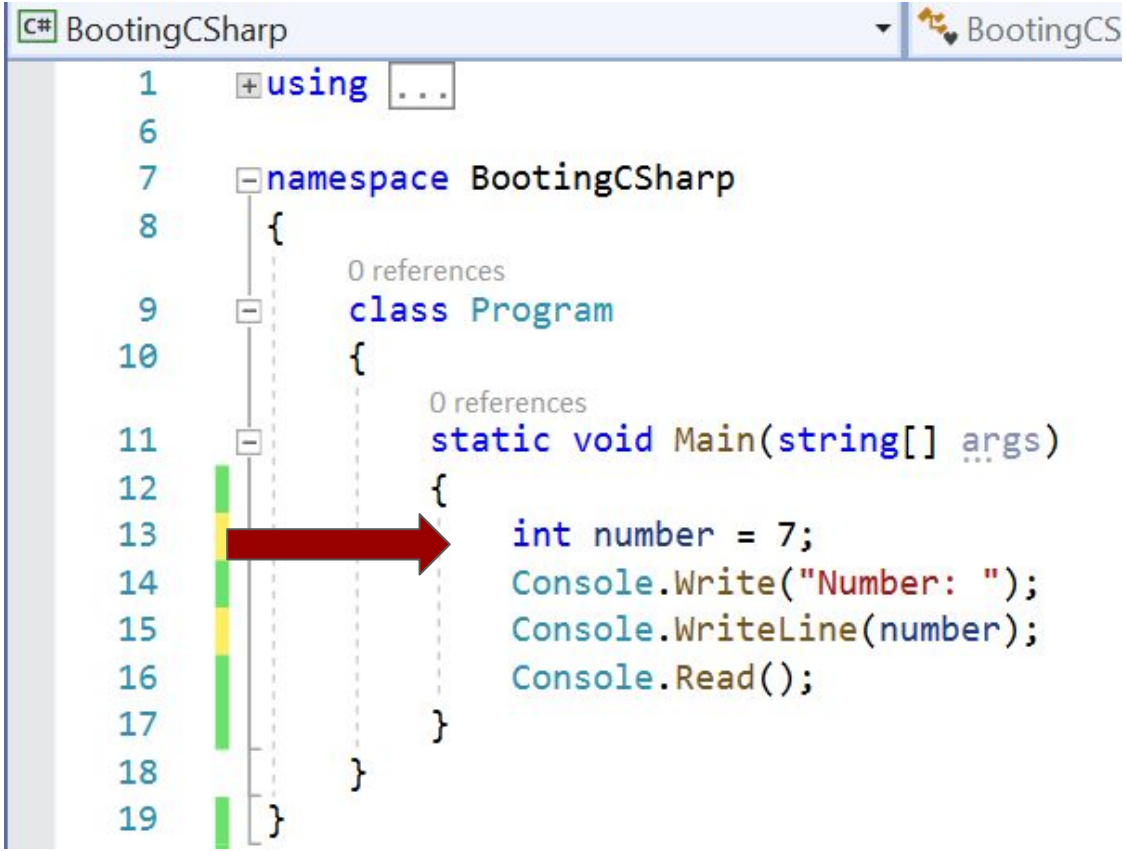
BootingCSharp | BootingCSharp.Program

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace BootingCSharp
8   {
        0 references
9       class Program
10      {
            0 references
11          static void Main(string[] args)
12          {
13              Console.WriteLine("Welcome to OOP Class");
14              Console.Read();
15          }
16      }
17  }
```

In C#,
Alternative to cout with endl is
Console.WriteLine(" ");

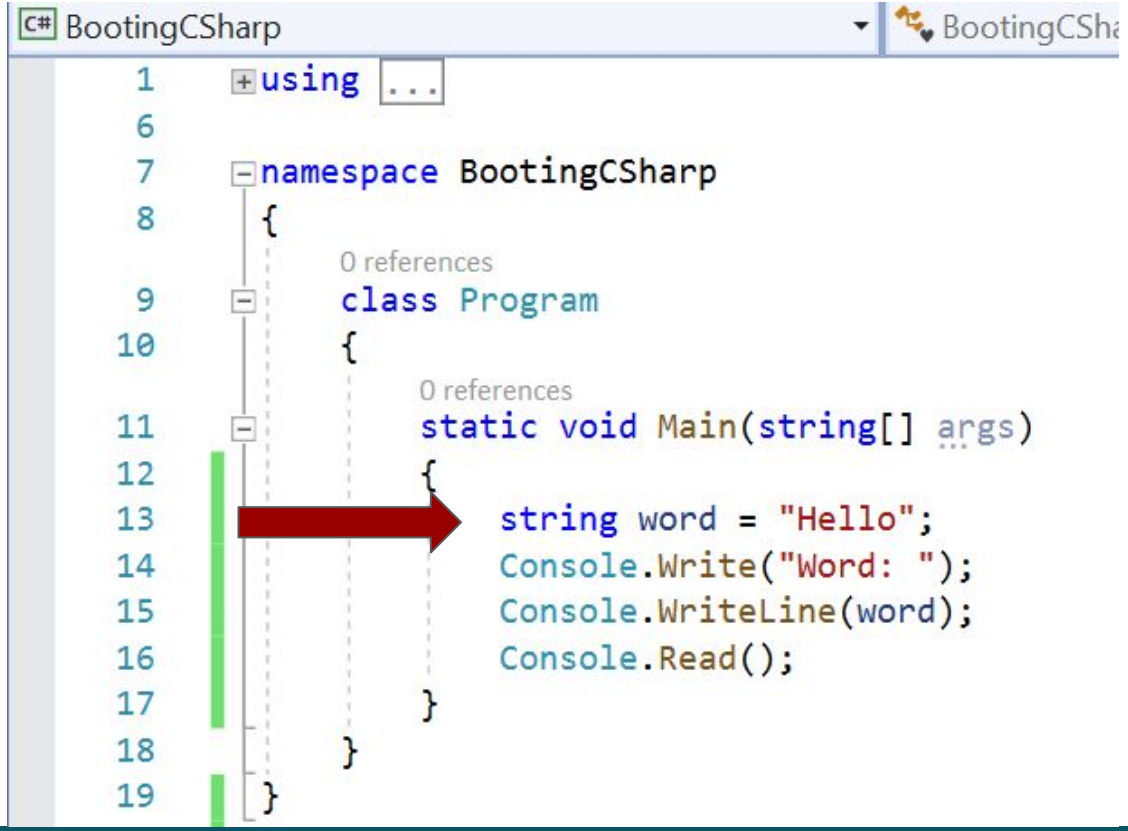
Declaring Variables in C#

Declaring Variables: int datatype



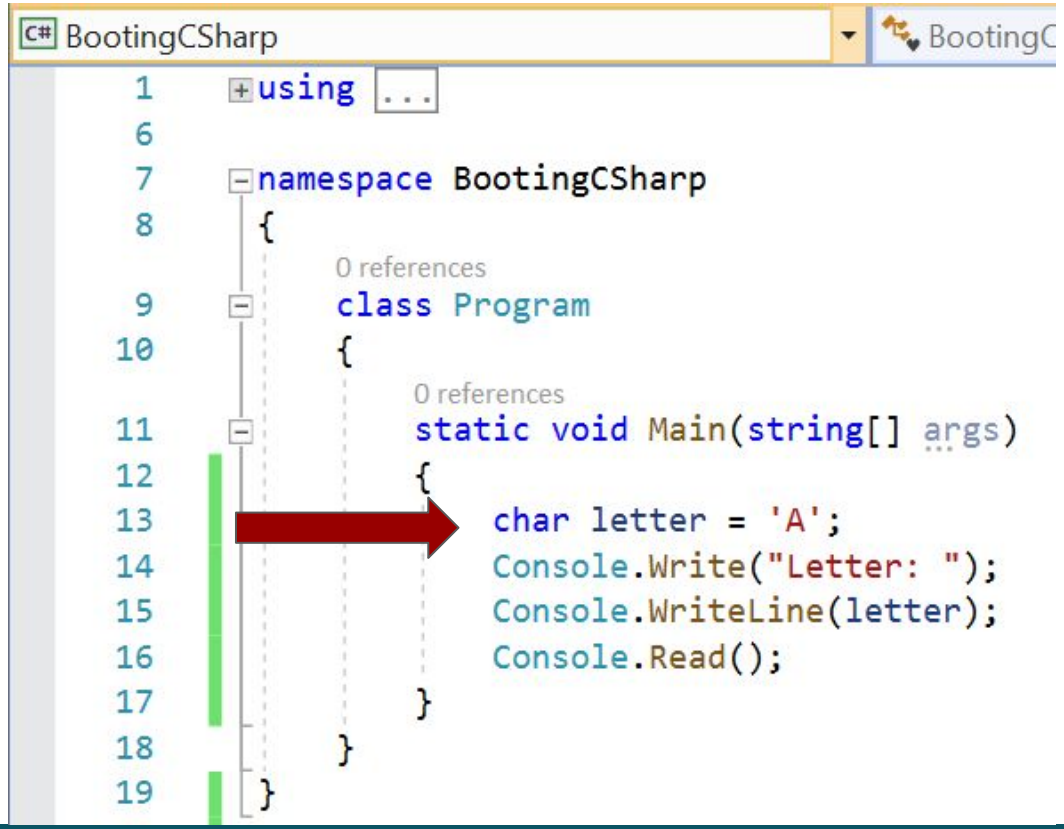
Same as in C++

Declaring Variables: string datatype



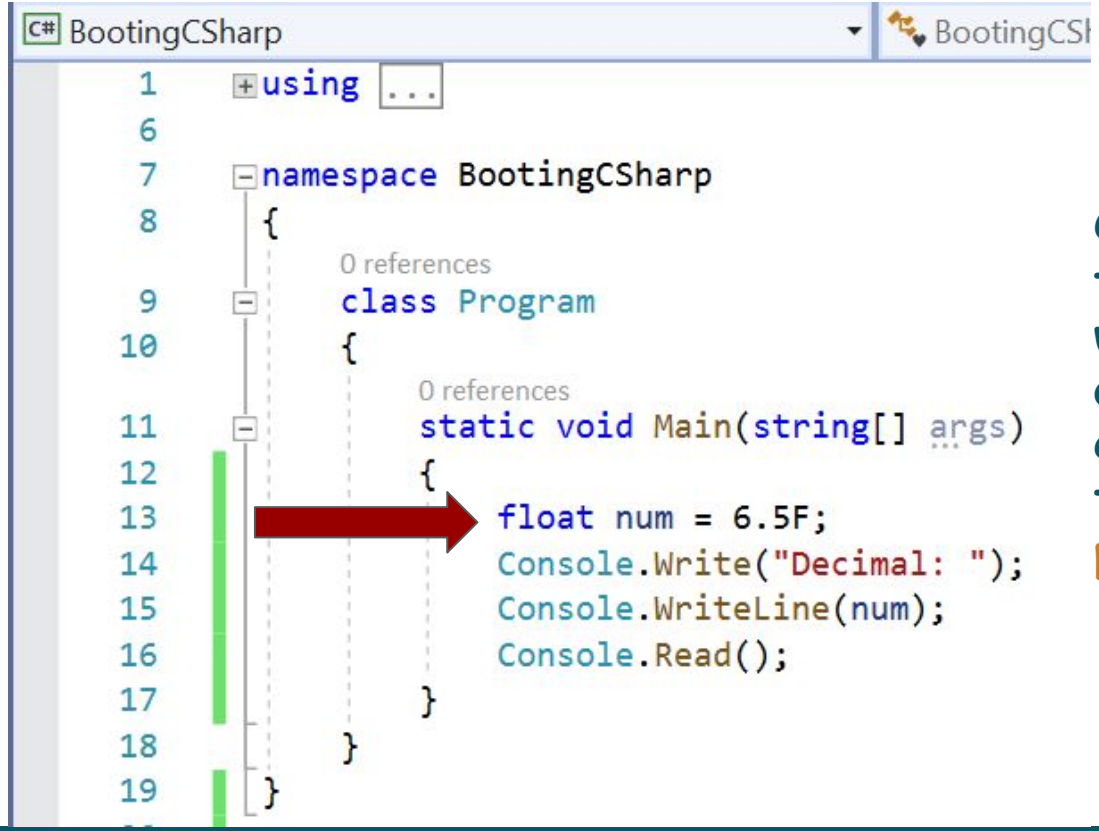
Same as in C++

Declaring Variables: char datatype



Same as in C++

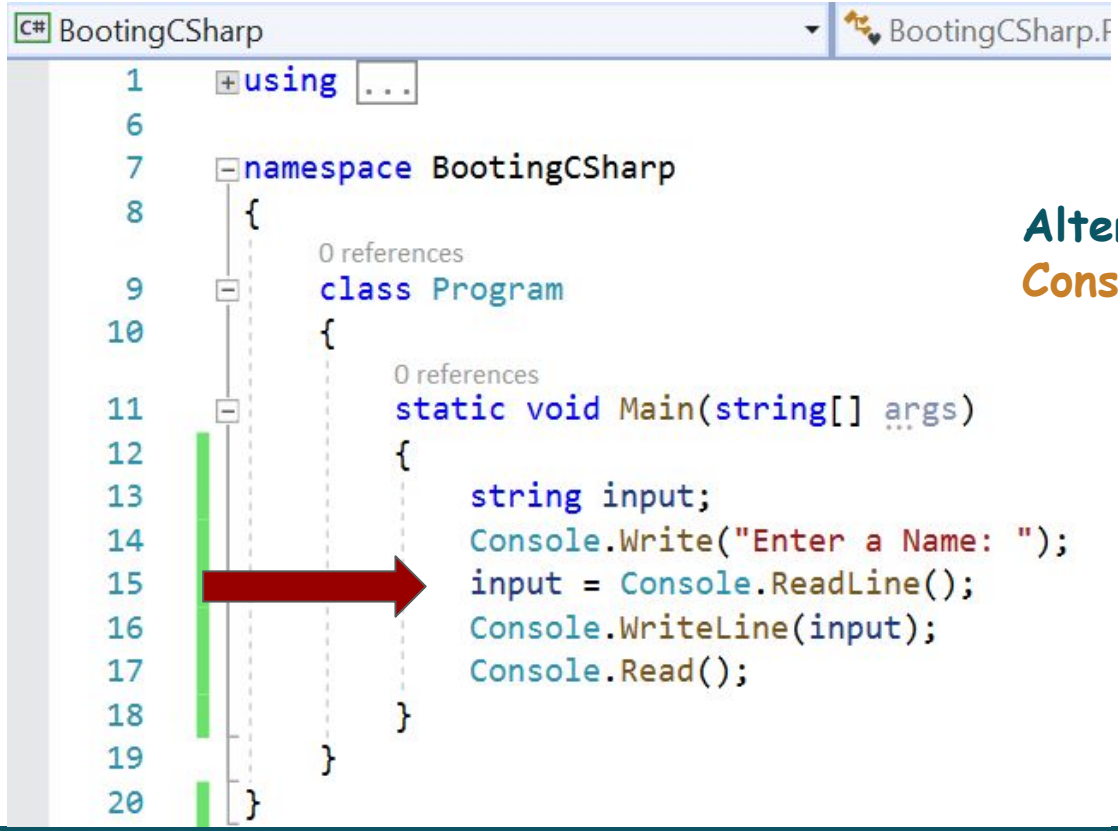
Declaring Variables: float datatype



Only difference is that you have to write F at the end of the number to explicitly mention that it is a floating point number

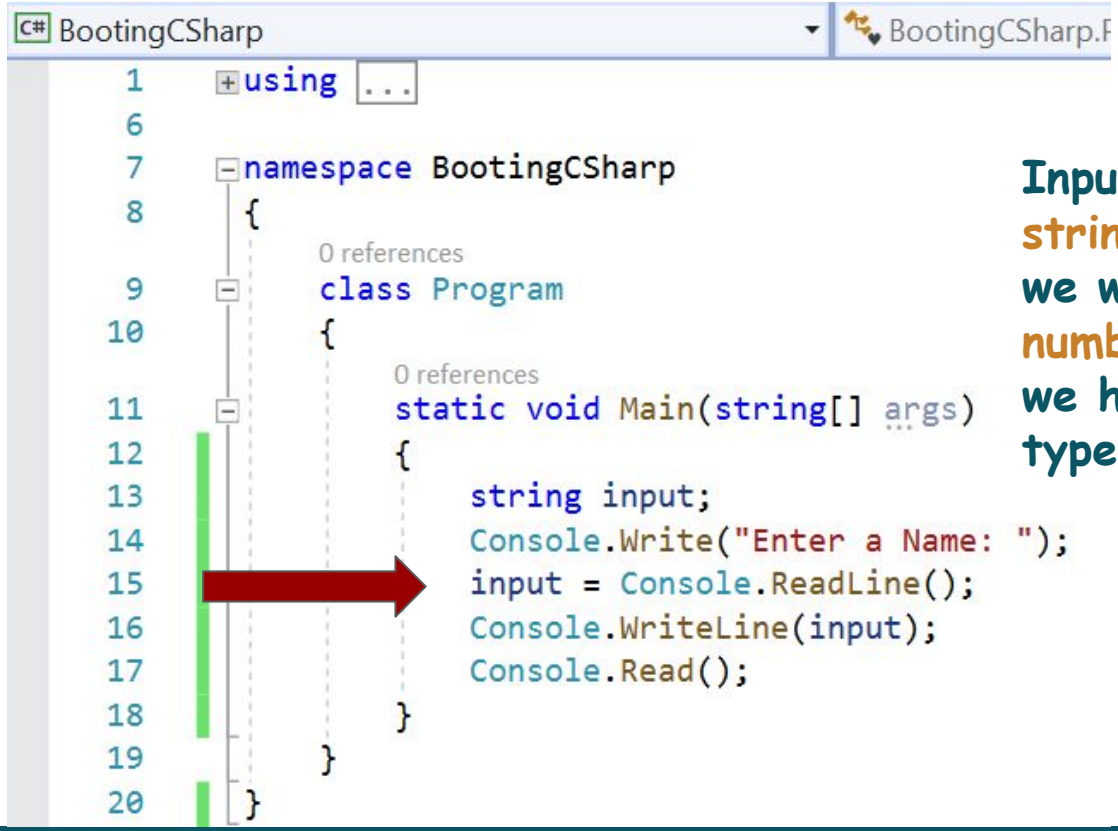
Taking Input in C#

Taking Input



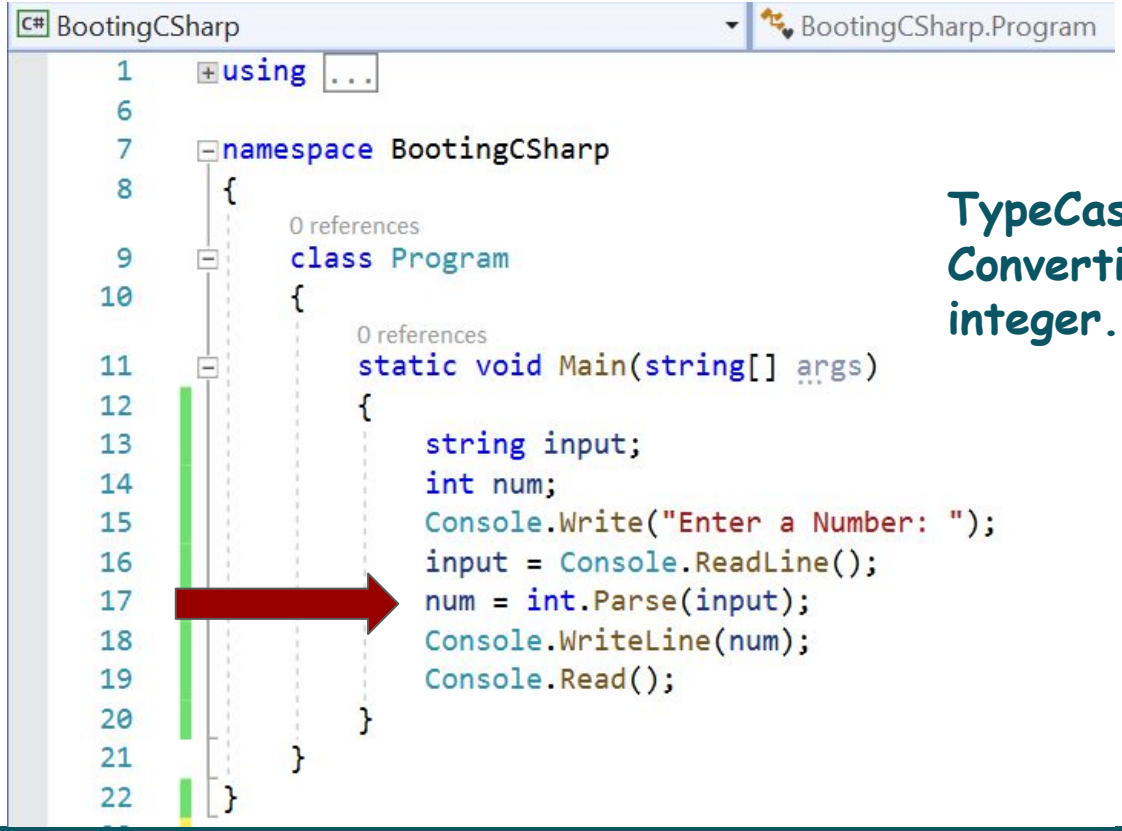
Alternative to cin is
Console.ReadLine();

Taking Input



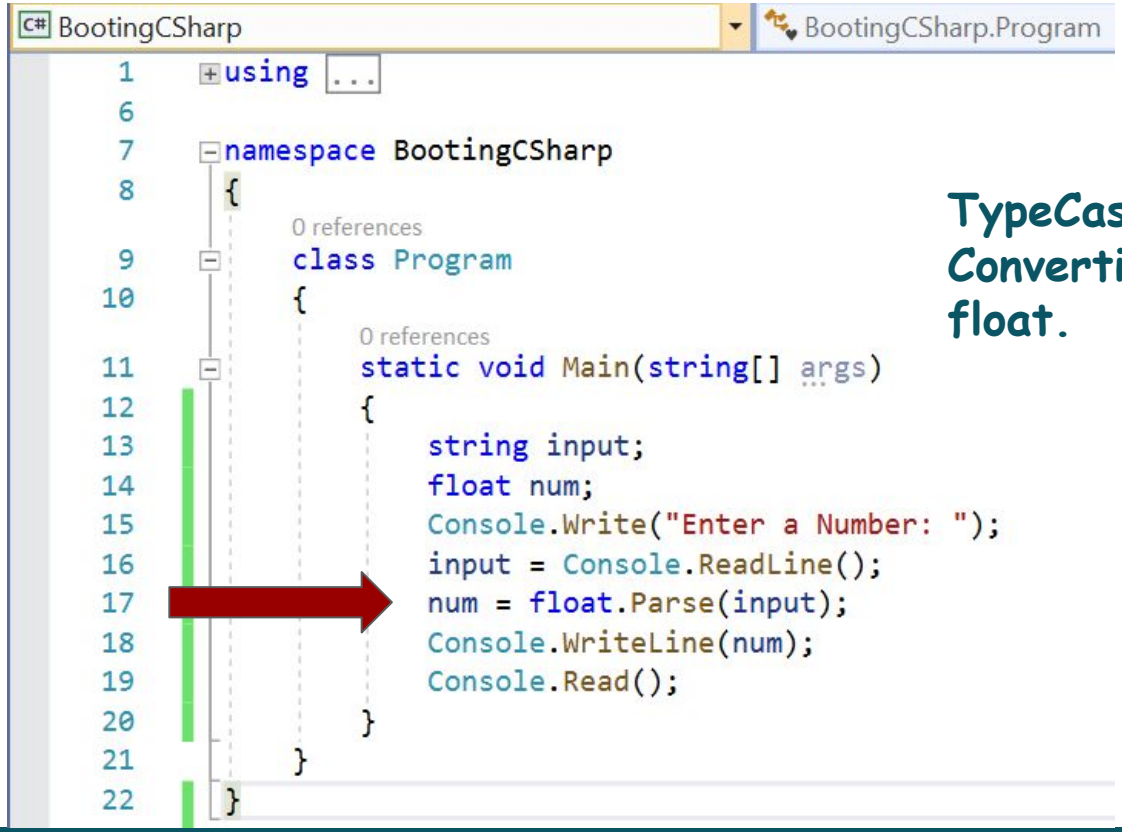
Input is always in a string. Therefore if we want to take a number as input then we have do typecasting.

Taking Input an int Number



TypeCasting,
Converting string into integer.

Taking Input a float Number



TypeCasting,
Converting string into
float.

Working Example: Vision

Write a program that takes length of side of a square as input and calculate its area using following formula:

Area= Length * Length

```
G:\OOP 2022\BootingCSharp\BootingCSharp\bin\Debug\BootingCSharp.exe
Enter the Length: 5
Area is:
25
```

Working Example: Solution

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        string input;
        float length;
        float area;
        Console.Write("Enter the Length: ");
        input = Console.ReadLine();
        length = float.Parse(input);
        area = length * length;
        Console.WriteLine("Area is: ");
        Console.WriteLine(area);
        Console.Read();
    }
}
```

Conditional Statements, Comparison and Logical Operators in C#

Working Example: Vision

We want to print "You are Passed" when Marks are greater than 50 but print "You are Failed" when Marks are less than or equal to 50.

Working Example: Solution

```
static void Main(string[] args)
{
    string input;
    float marks;
    Console.Write("Enter the Marks: ");
    input = Console.ReadLine();
    marks = float.Parse(input);
    if (marks > 50)
    {
        Console.WriteLine("You are Passed");
    }
    else
    {
        Console.WriteLine("You are Failed");
    }
    Console.Read();
}
```

Same as in C++

Loops in C#

Loops in C#

We have two type of loops

- Counter Loops
- Conditional Loops

Counter Loop

Count the steps and execute until specific number of times.

Condition Loop

Execute the steps until a specific condition is not met.

Working Example: Vision

We want to print "Welcome Jack" 5 times.

For Loop: Solution

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

    for (int x = 0; x < 5; x++)
    {
        Console.WriteLine("Welcome Jack");
    }
    Console.Read();
}
```

Same as in C++

Working Example: Vision

Write a program that keeps taking input from the user and sum up all these input until he enters -1. When user enters -1 the program should print sum of all values.

While Loop: Solution

```
static void Main(string[] args)
{
    int num;
    int sum = 0;
    Console.Write("Enter Number: ");
    num = int.Parse(Console.ReadLine());
    while (num != -1)
    {
        sum = sum + num;
        Console.Write("Enter Number: ");
        num = int.Parse(Console.ReadLine());
    }
    Console.WriteLine("The total sum is {0}", sum);
    Console.Read();
}
```

Same as in C++

Do While Loop: Solution

```
static void Main(string[] args)
{
    int num;
    int sum = 0;
    do
    {
        Console.Write("Enter Number: ");
        num = int.Parse(Console.ReadLine());
        sum = sum + num;
    }
    while (num != -1);
    sum = sum + 1;
    Console.WriteLine("The total sum is {0}", sum);
    Console.Read();
}
```

Same as in C++

Working Example: Vision

Write a program that takes 3 numbers as input and print the largest of them.

Solution with Variables

Why do not we declare multiple variables and use a for loop ?

```
int fnum, snum, tnum;
Console.Write("Enter First Numbers:");
fnum = int.Parse(Console.ReadLine());
Console.Write("Enter Secon Numbers:");
snum = int.Parse(Console.ReadLine());
Console.Write("Enter Third Numbers:");
tnum = int.Parse(Console.ReadLine());
if (fnum > snum && fnum > tnum){
    Console.WriteLine("First Number is Greater");
}
if (snum > fnum && snum > tnum)                {
    Console.WriteLine("Second Number is Greater");
}
if (tnum > fnum && tnum > snum)                {
    Console.WriteLine("Third Number is Greater");
}
```

Solution with Variables

It would be easier if we have same variable name for same type of data and if it require to access any data we just give its number

```
int fnum, snum, tnum;
Console.Write("Enter First Numbers:");
fnum = int.Parse(Console.ReadLine());
Console.Write("Enter Secon Numbers:");
snum = int.Parse(Console.ReadLine());
Console.Write("Enter Third Numbers:");
tnum = int.Parse(Console.ReadLine());
if (fnum > snum && fnum > tnum){
    Console.WriteLine("First Number is Greater");
}
if (snum > fnum && snum > tnum)                {
    Console.WriteLine("Second Number is Greater");
}
if (tnum > fnum && tnum > snum)                {
    Console.WriteLine("Third Number is Greater");
}
```

Arrays in C#

Arrays

You can access variables through same name with different index value [0]

```
int[] value = new int[5];
```

Solution with Arrays

```
static void Main(string[] args)
{
    //Taking input
    int[] numbers=new int[3];
    for (int idx = 0; idx < 3; idx++) {
        Console.Write("Enter the Number {0}:",idx);
        numbers[idx] = int.Parse(Console.ReadLine());
    }
```

```
int fnum, snum, tnum;
Console.Write("Enter First Numbers:");
fnum = int.Parse(Console.ReadLine());
Console.Write("Enter Secon Numbers:");
snum = int.Parse(Console.ReadLine());
Console.Write("Enter Third Numbers:");
tnum = int.Parse(Console.ReadLine());
if (fnum > snum && fnum > tnum){
    Console.WriteLine("First Number is Greater");
}
if (snum > fnum && snum > tnum)            {
    Console.WriteLine("Second Number is Greater");
}
if (tnum > fnum && tnum > snum)            {
    Console.WriteLine("Third Number is Greater");
}
```

Solution with Arrays

```
static void Main(string[] args)
{
    //Taking input
    int[] numbers=new int[3];
    for (int idx = 0; idx < 3; idx++) {
        Console.Write("Enter the Number {0}:",idx);
        numbers[idx] = int.Parse(Console.ReadLine());
    }
    //Finding the Largest
    int largest = -1;
    for (int idx = 0; idx < 3; idx++) {
        if (numbers[idx] > largest) {
            largest = numbers[idx];
        }
    }
    Console.WriteLine("Largest is: {0}", largest);
    Console.Read();
}
```

```
int fnum, snum, tnum;
Console.Write("Enter First Numbers:");
fnum = int.Parse(Console.ReadLine());
Console.Write("Enter Secon Numbers:");
snum = int.Parse(Console.ReadLine());
Console.Write("Enter Third Numbers:");
tnum = int.Parse(Console.ReadLine());
if (fnum > snum && fnum > tnum){
    Console.WriteLine("First Number is Greater");
}
if (snum > fnum && snum > tnum)             {
    Console.WriteLine("Second Number is Greater");
}
if (tnum > fnum && tnum > snum)             {
    Console.WriteLine("Third Number is Greater");
}
```

Learning Objective

Know the programming constructs of C# and Solve Complex problems in C# language.

Self Assessment

1. Lilly is N years old. For each birthday she receives a present. For each odd birthday (1, 3, 5, …, n) she receives toys for each even birthday (2, 4, 6, …, n) she receives money. For her second birthday she received 10.00 USD, and the amount is increased by 10.00 USD for each following even birthday (2 -> 10, 4 -> 20, 6 -> 30 etc.).
Over the years Lilly has secretly saved her money. Lilly's brother, in the years when she received money, took 1.00 USD from each of the amounts.
Lilly has sold the toys, received over the years, each one for P USD and added the sum to the amount of saved money.
With the money she wanted to buy a washing machine for X USD.
Write a Function that takes lily's age, price of washing machine and price of toy and calculates how much money she has saved and if it is enough to buy a washing machine.

Self Assessment

Input:

We read from the console 3 numbers, each on a separate line:

1. Lilly's age – integer in the range of [1 … 77].
2. Price of the washing machine – number in the range of [1.00 … 10,000.00].
3. Unit price of each toy – integer in the range of [0 … 40].

Output

Print on the console one single line:

1. If Lilly's money is enough:
   "Yes! {N}" – where N is the remaining money after the purchase
2. If the money is not enough:

"No! {M}" – where M is the insufficiency amount

Self Assessment

Input	Output	Comments
10 170.00 6	Yes! 5.00	For the first birthday she gets a toy; 2nd -> 10 USD; 3rd -> toy; 4th -> 10 + 10 = 20 USD; 5th -> toy; 6th -> 20 + 10 = 30 USD; 7th -> toy; 8th -> 30 + 10 = 40 USD; 9th -> toy; 10th -> 40 + 10 = 50 USD. She has saved: 10 + 20 + 30 + 40 + 50 = 150 USD. She sold 5 toys for 6 USD each = 30 USD. Her brother took 1 USD 5 times = 5 USD. Remaining amount: 150 + 30 – 5 = 175 USD. 175 >= 170 (price of the washing machine): she managed to buy it and is left with 175-170 = 5 USD.
21 1570.98 3	No! 997.98	She has saved 550 USD. She has sold 11 toys, 3 USD each = 33 USD. Her brother has taken for 10 years 1 USD each year = 10 USD. Remaining amount: 550 + 33 - 10 = 573 USD. 573 < 1570.98: she did not manage to buy a washing machine. The insufficiency amount is: 1570.98 - 573 = 997.98 USD.