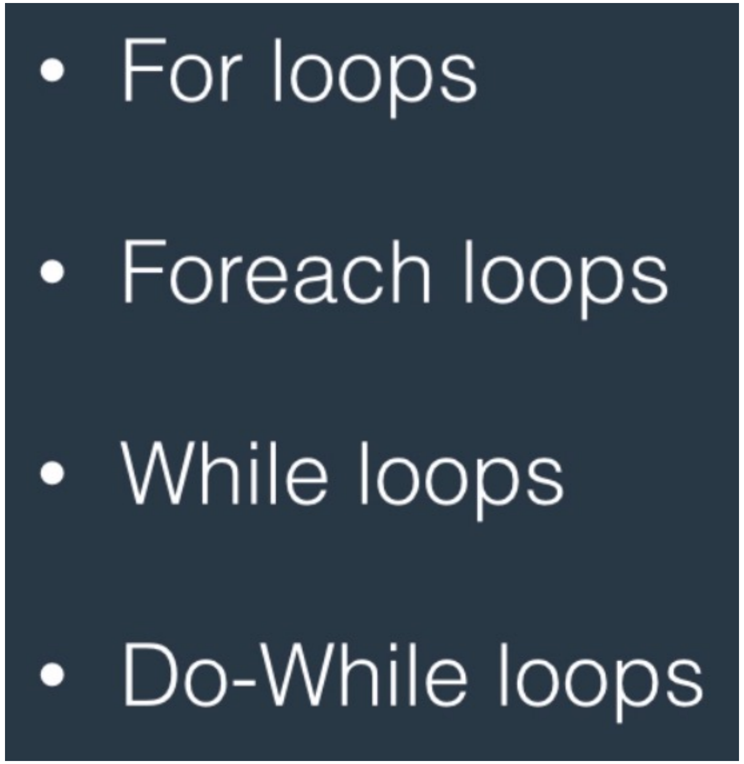
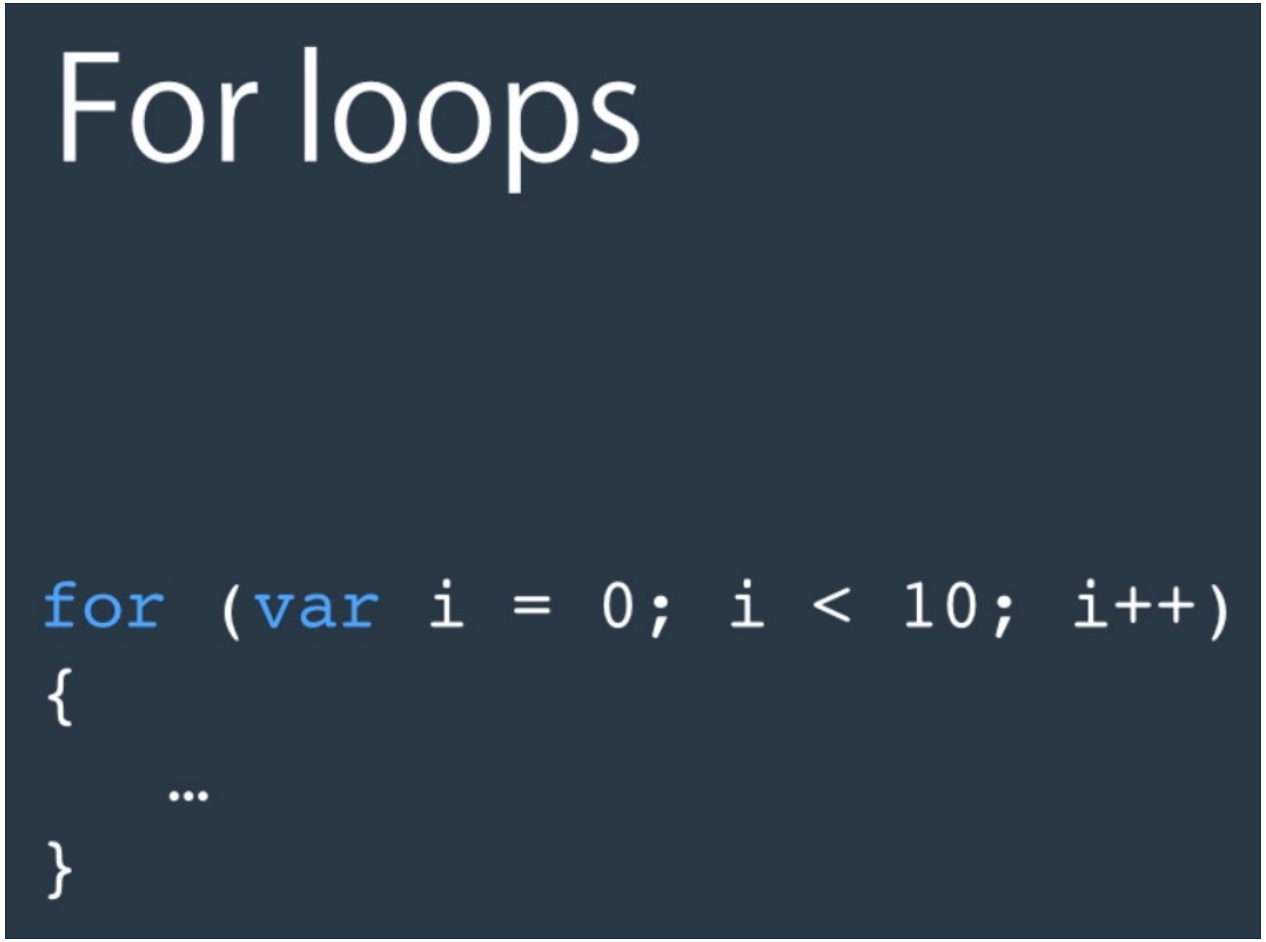
Iteration Statments

C# Iteration Statements

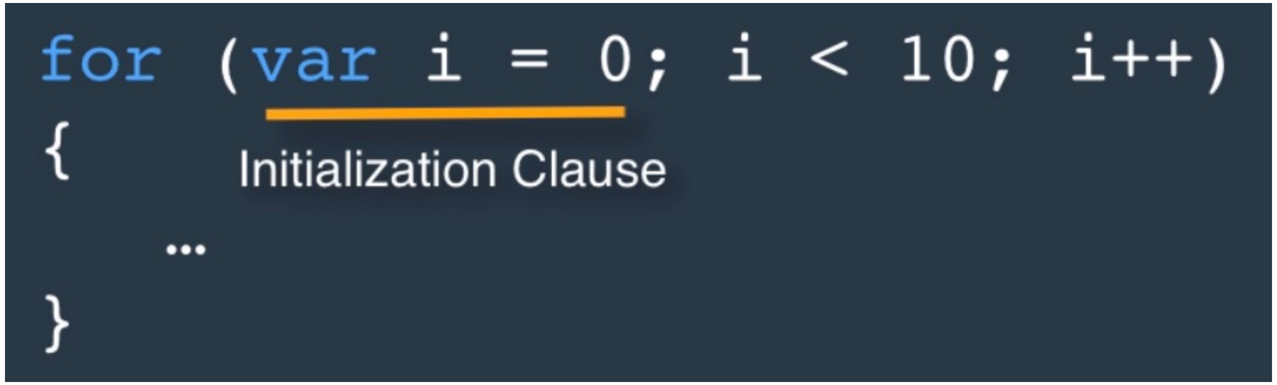


- For loops
- Foreach loops
- While loops
- Do-While loops



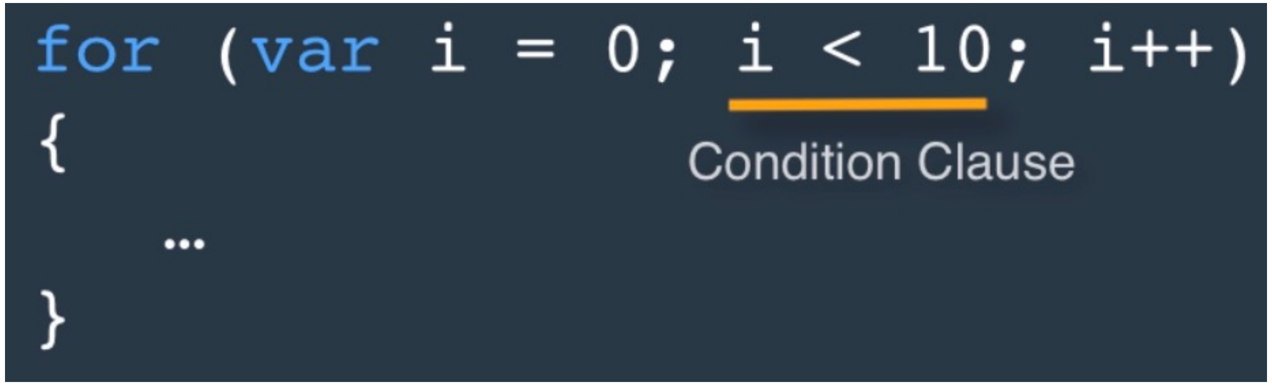
For loops

```
for (var i = 0; i < 10; i++)
{
    …
}
```



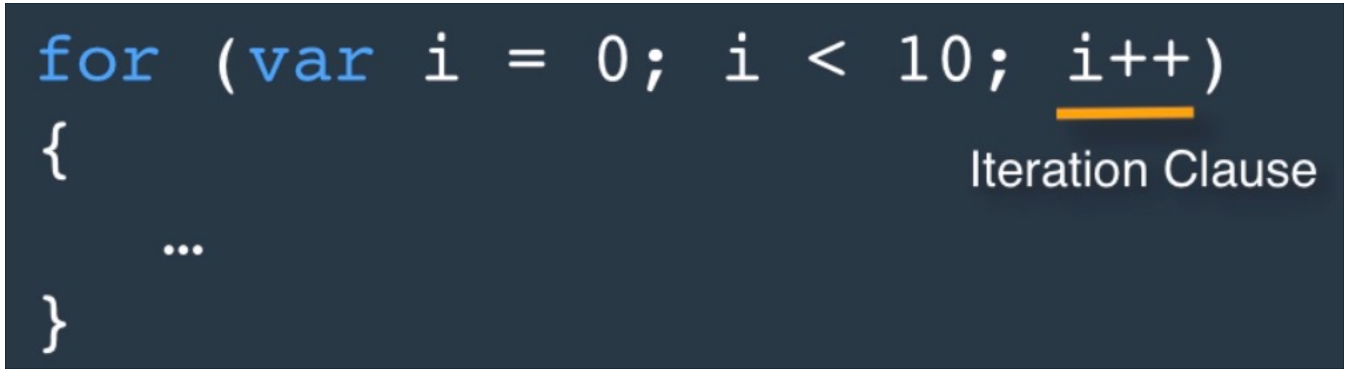
```
for (var i = 0; i < 10; i++)
{
    ...
}
```

Initialization Clause



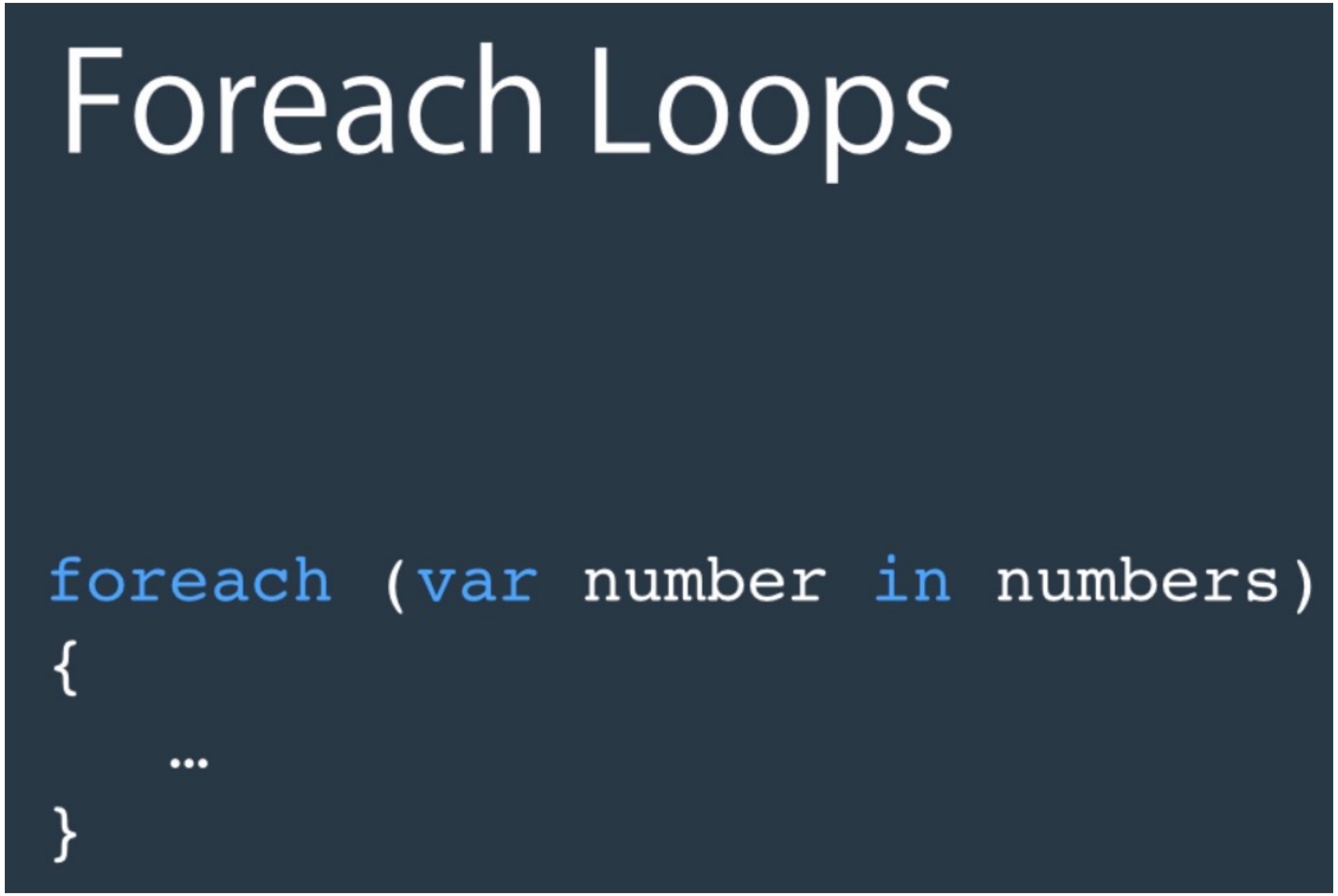
```
for (var i = 0; i < 10; i++)
{
    ...
}
```

Condition Clause



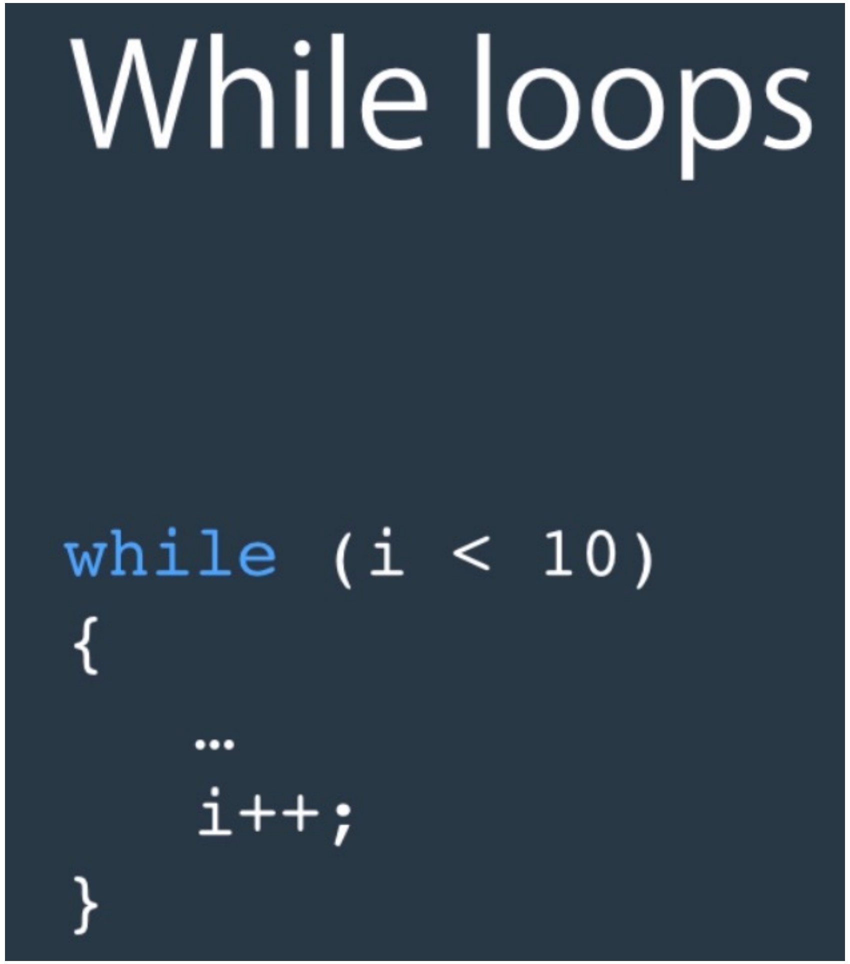
```
for (var i = 0; i < 10; i++)
{
    ...
}
```

Iteration Clause



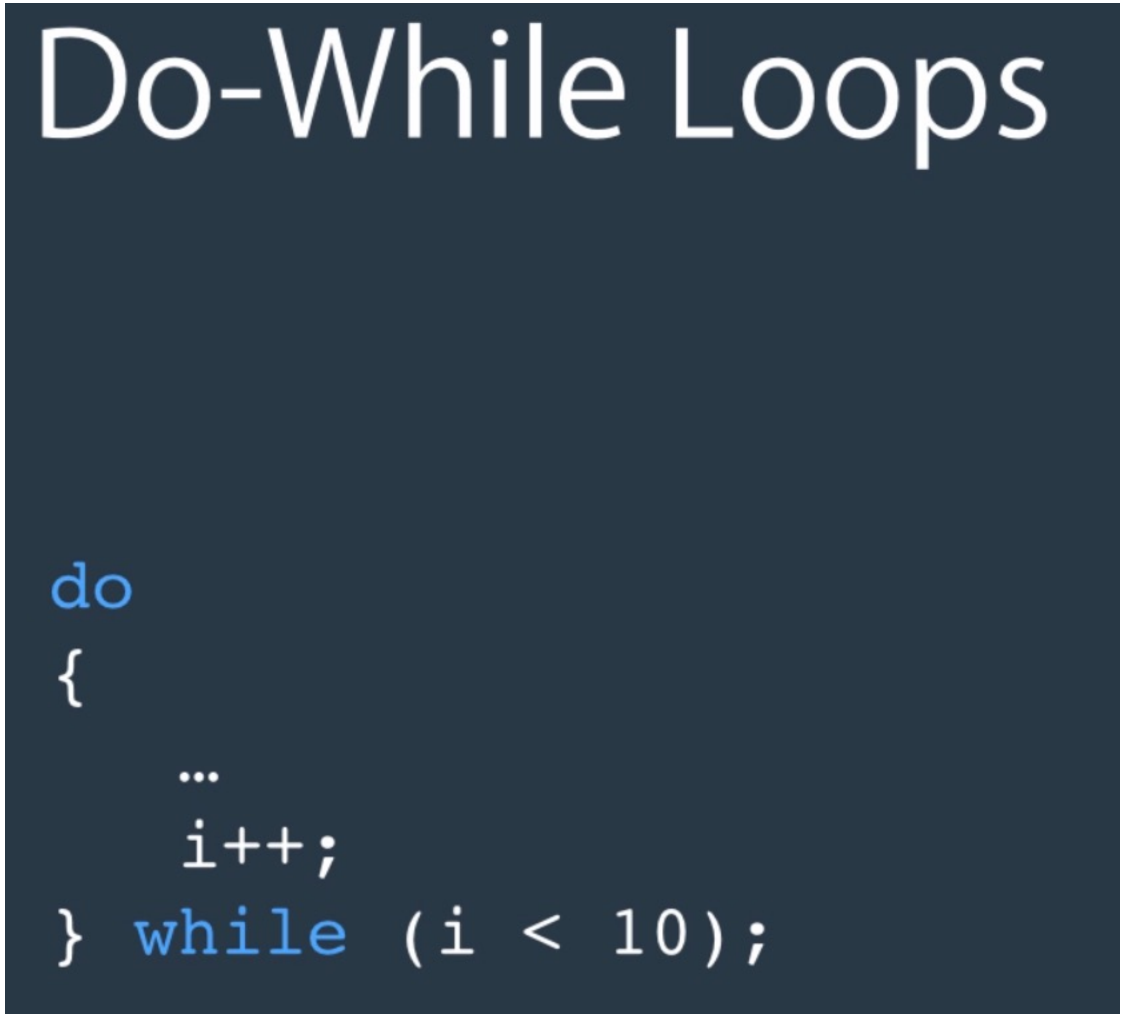
Foreach Loops

```
foreach (var number in numbers)
{
    …
}
```



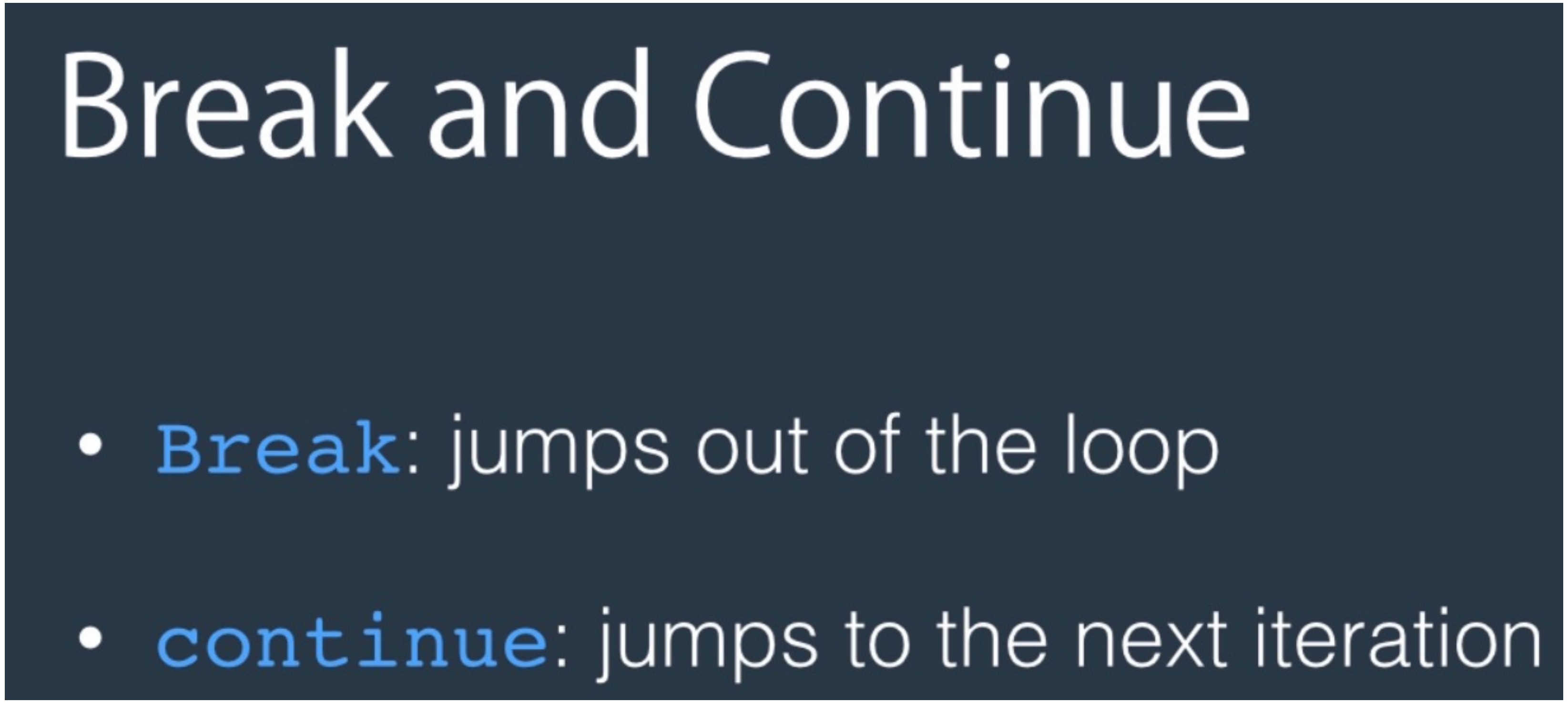
While loops

```
while (i < 10)
{
    …
    i++;
}
```



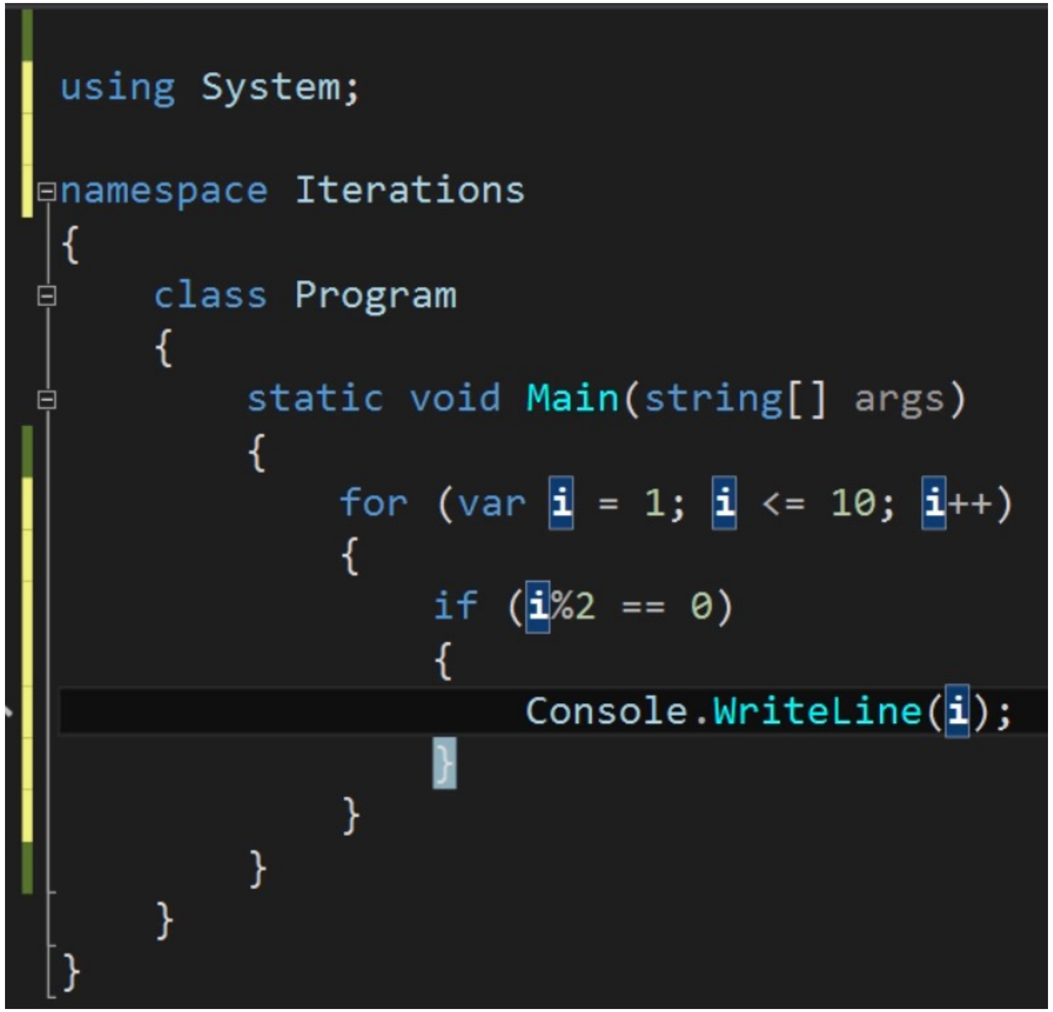
Do-While Loops

```
do
{
    …
    i++;
} while (i < 10);
```

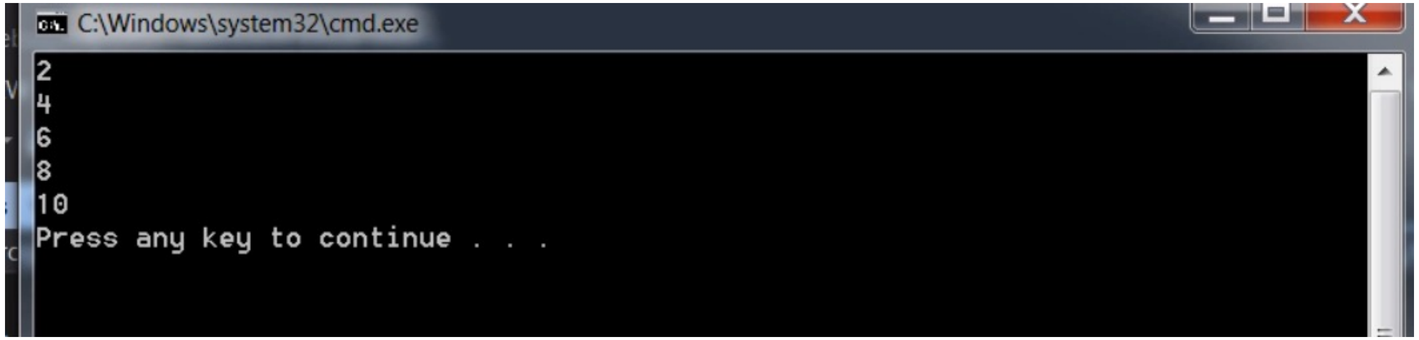


Break and Continue

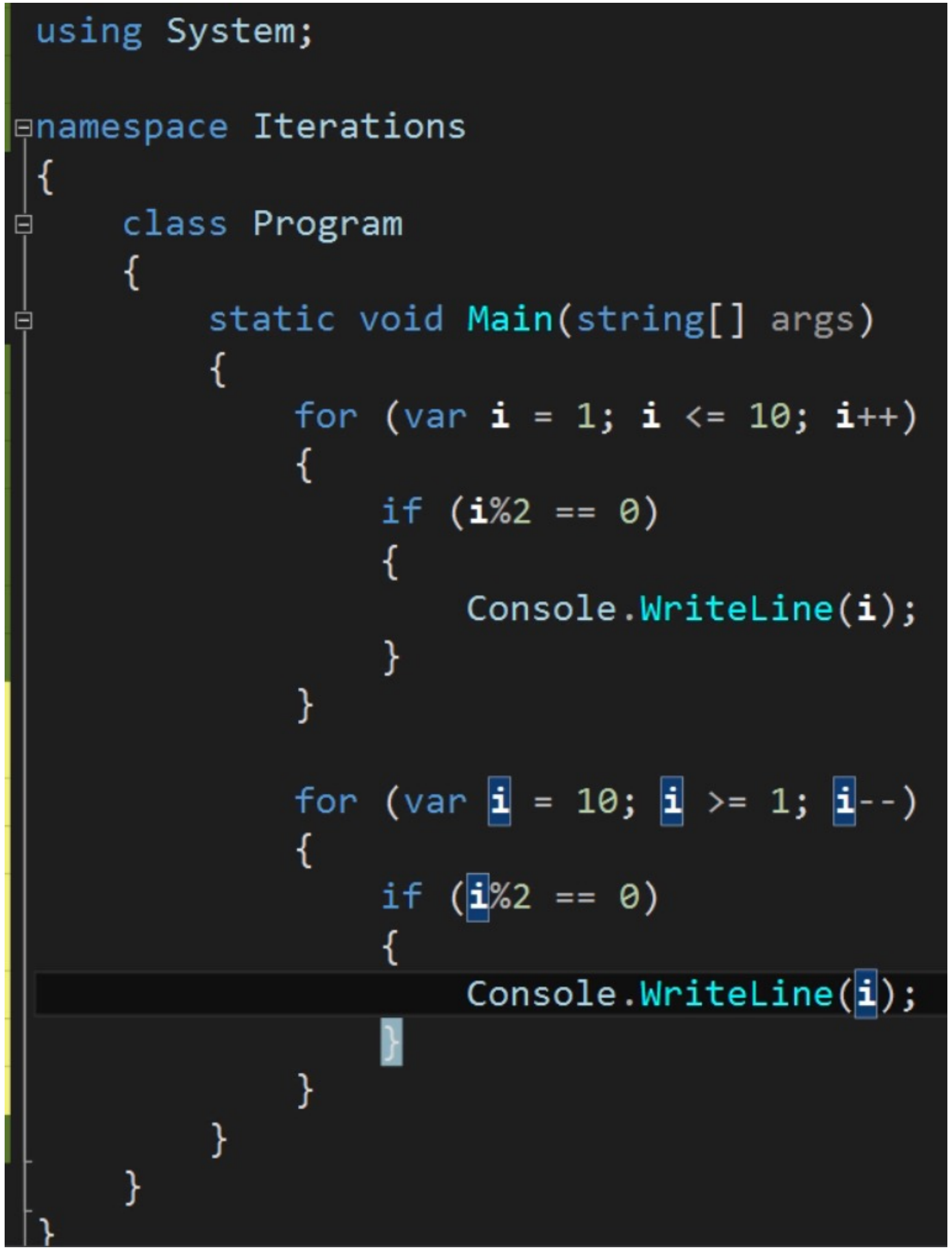
- `Break`: jumps out of the loop
- `continue`: jumps to the next iteration



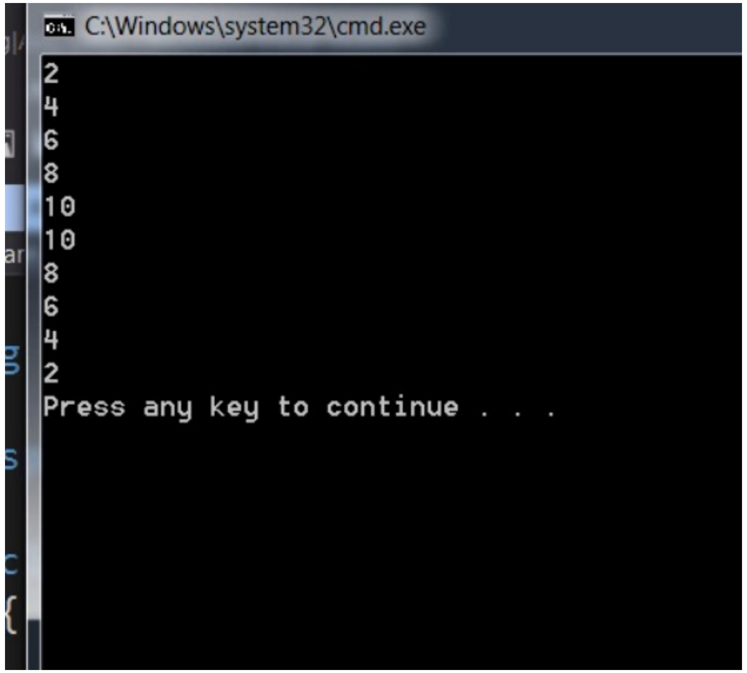
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            for (var i = 1; i <= 10; i++)
            {
                if (i%2 == 0)
                {
                    Console.WriteLine(i);
                }
            }
        }
    }
}
```



```
2
4
6
8
10
Press any key to continue . . .
```

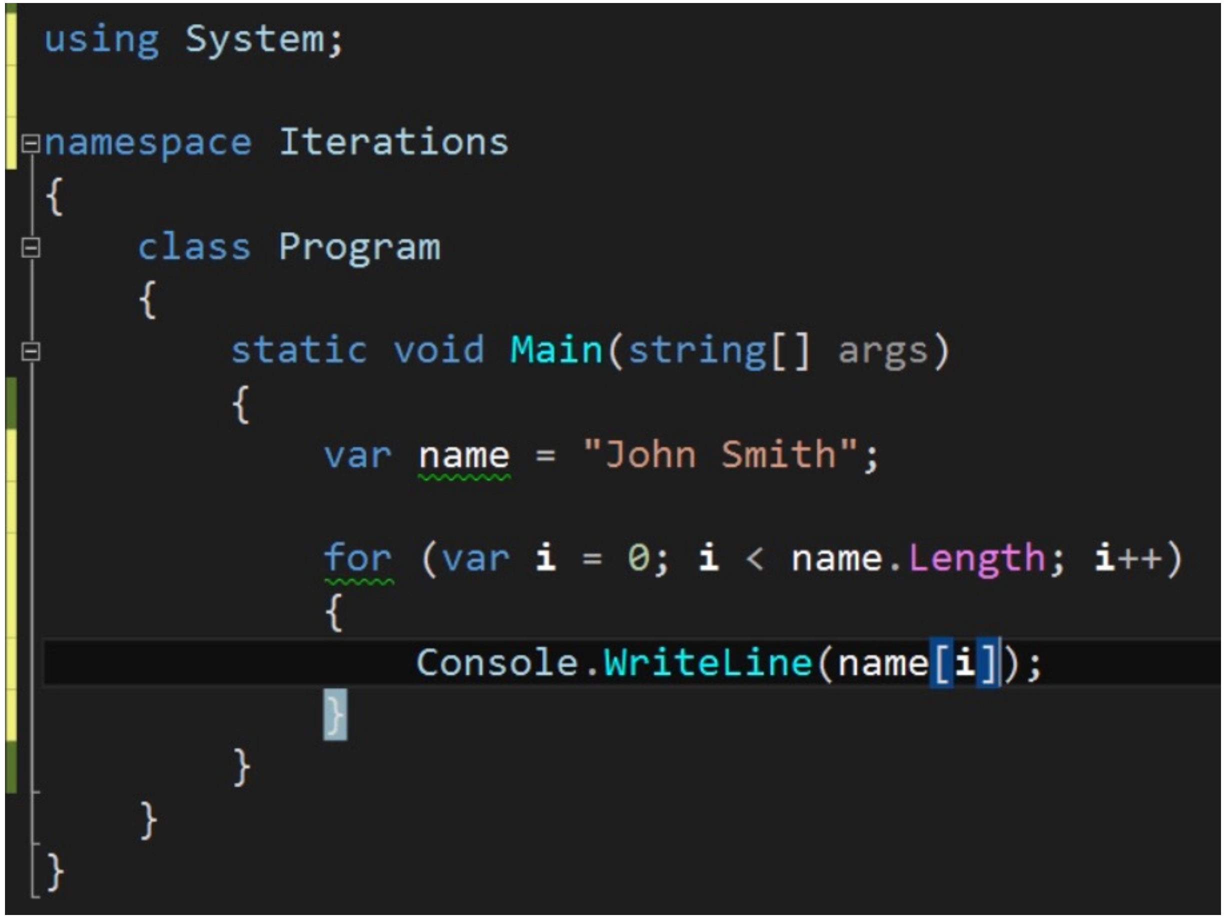


```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            for (var i = 1; i <= 10; i++)
            {
                if (i%2 == 0)
                {
                    Console.WriteLine(i);
                }
            }

            for (var i = 10; i >= 1; i--)
            {
                if (i%2 == 0)
                {
                    Console.WriteLine(i);
                }
            }
        }
    }
}
```

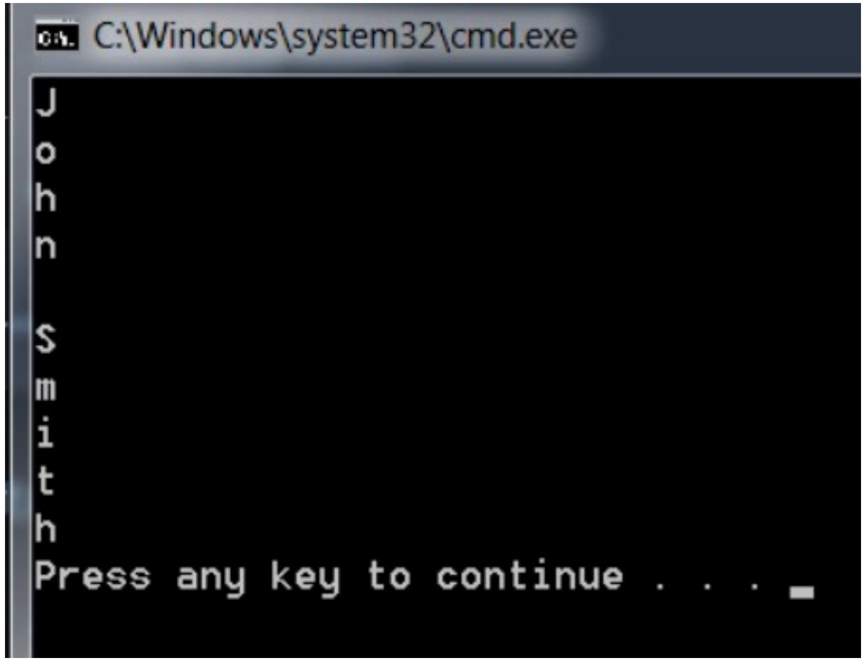


C:\Windows\system32\cmd.exe

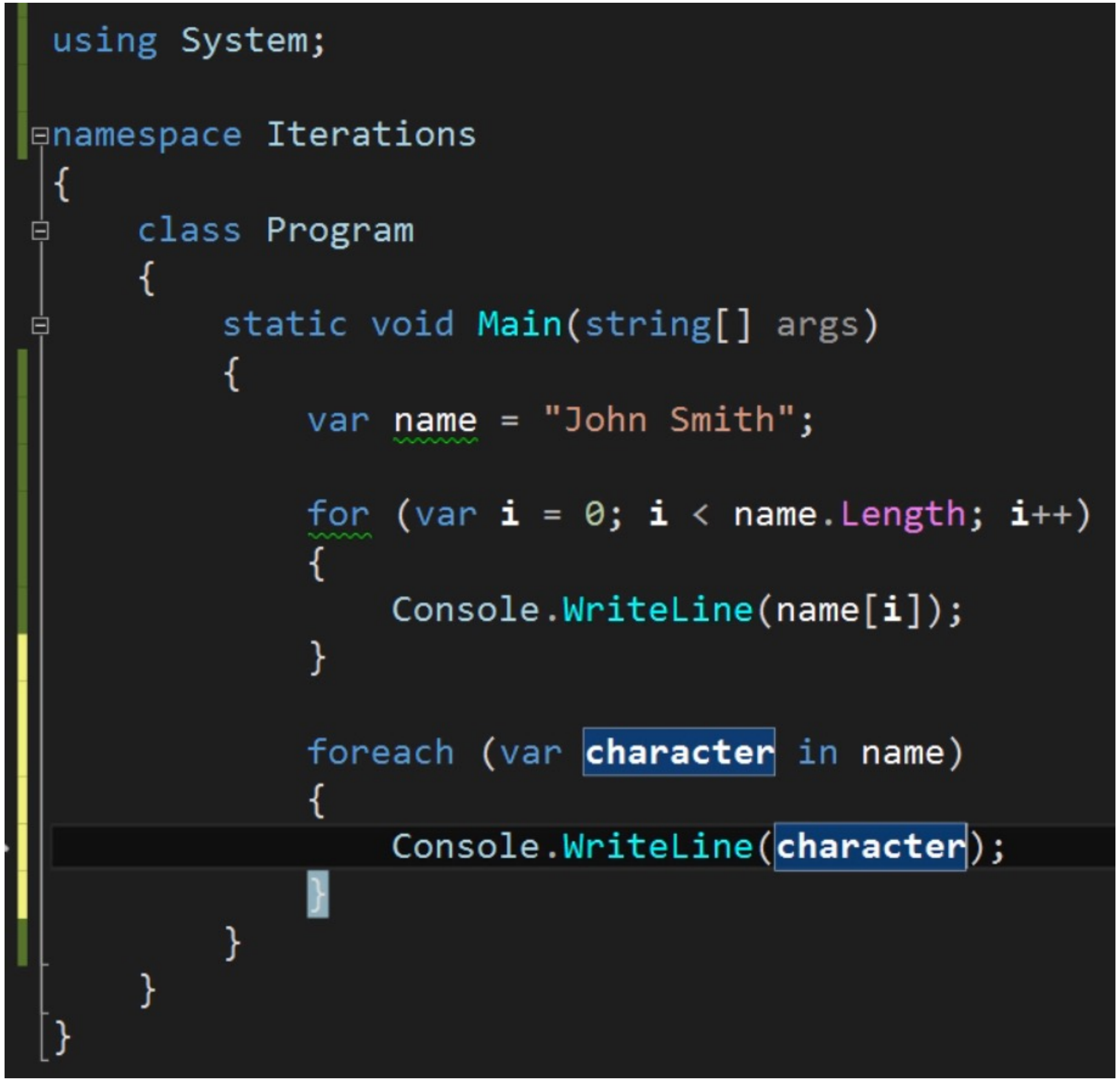
```
2
4
6
8
10
10
8
6
4
2
Press any key to continue . . .
```



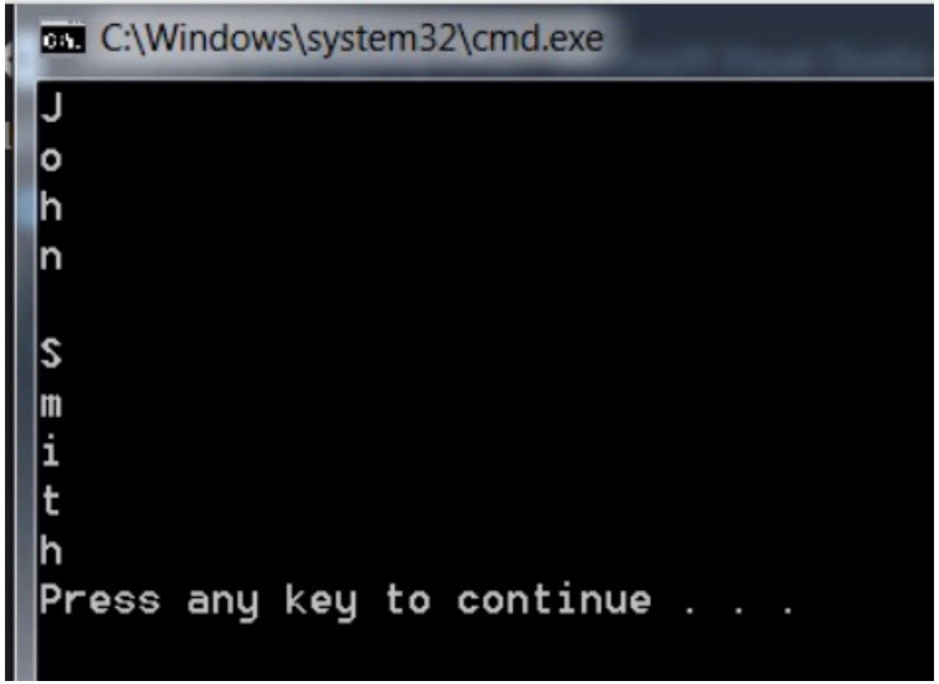
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            var name = "John Smith";

            for (var i = 0; i < name.Length; i++)
            {
                Console.WriteLine(name[i]);
            }
        }
    }
}
```



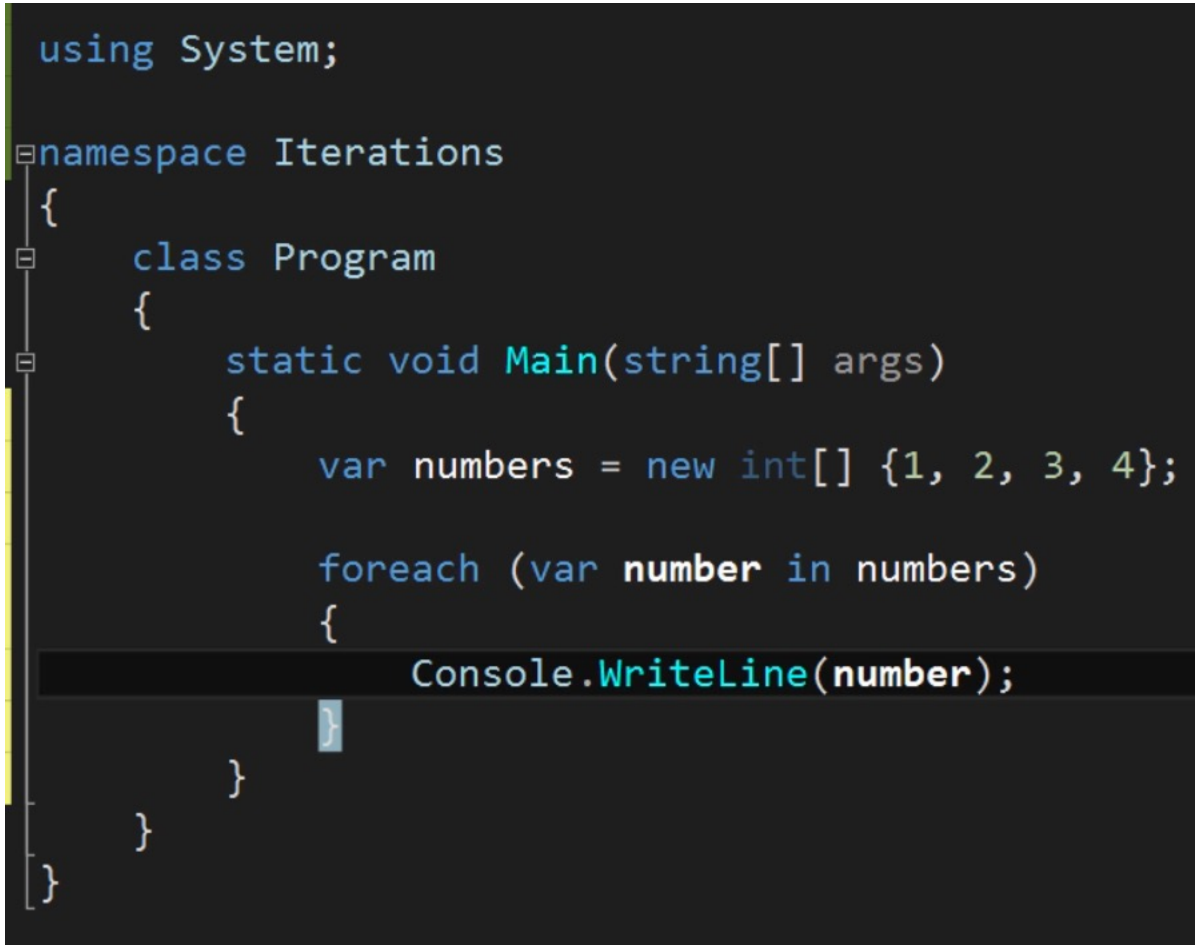
```
J
o
h
n

S
m
i
t
h
Press any key to continue . . .
```



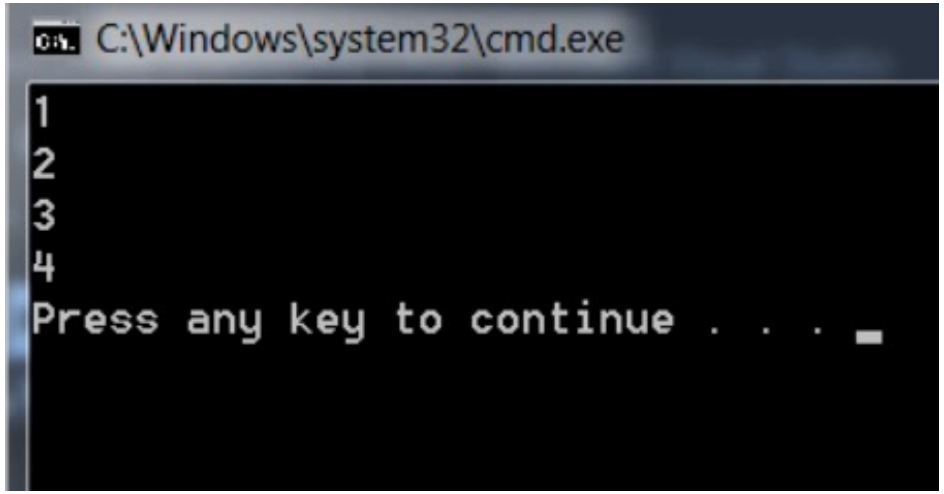
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            var name = "John Smith";

            for (var i = 0; i < name.Length; i++)
            {
                Console.WriteLine(name[i]);
            }

            foreach (var character in name)
            {
                Console.WriteLine(character);
            }
        }
    }
}
```



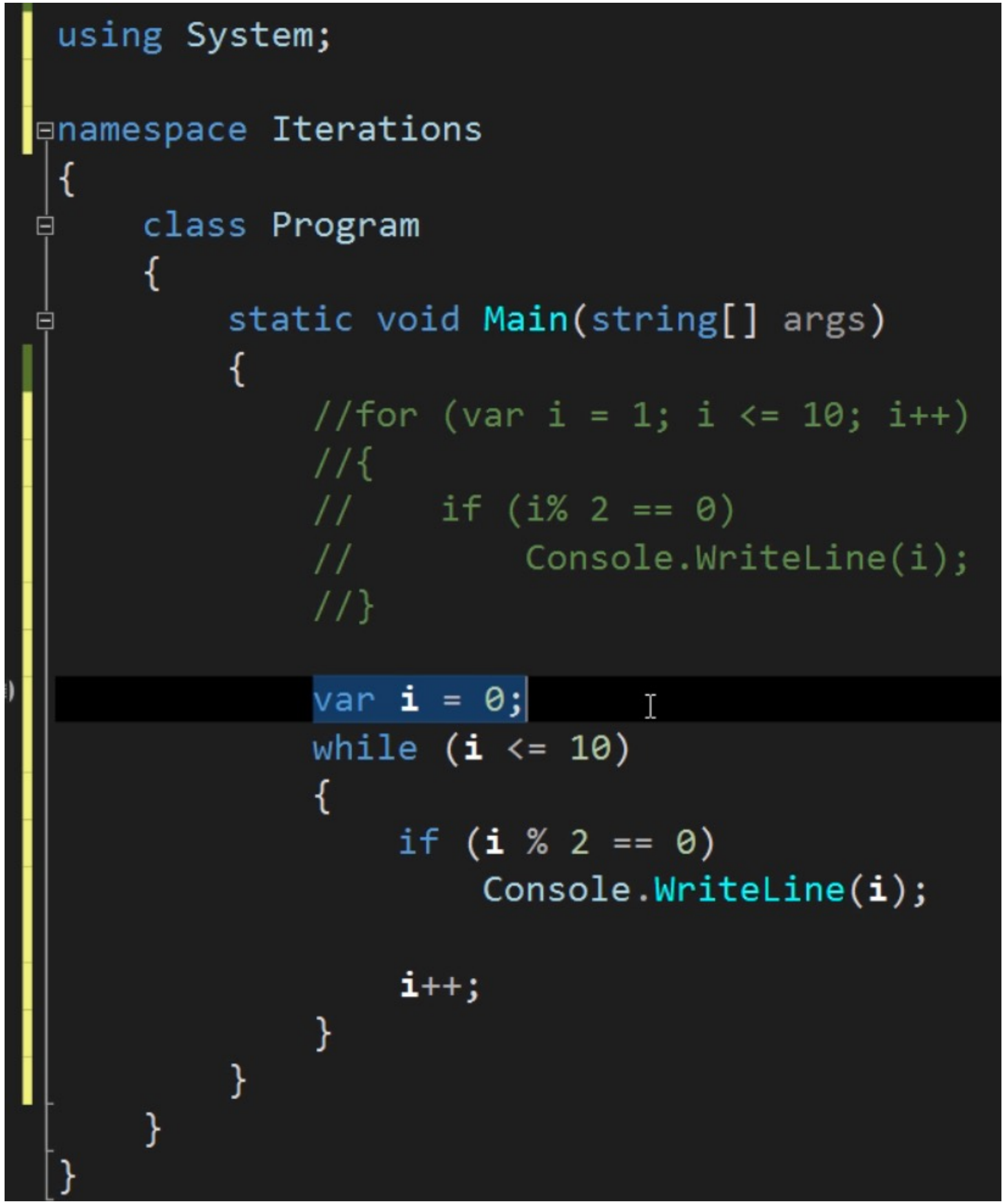
```
J
o
h
n

S
m
i
t
h
Press any key to continue . . .
```



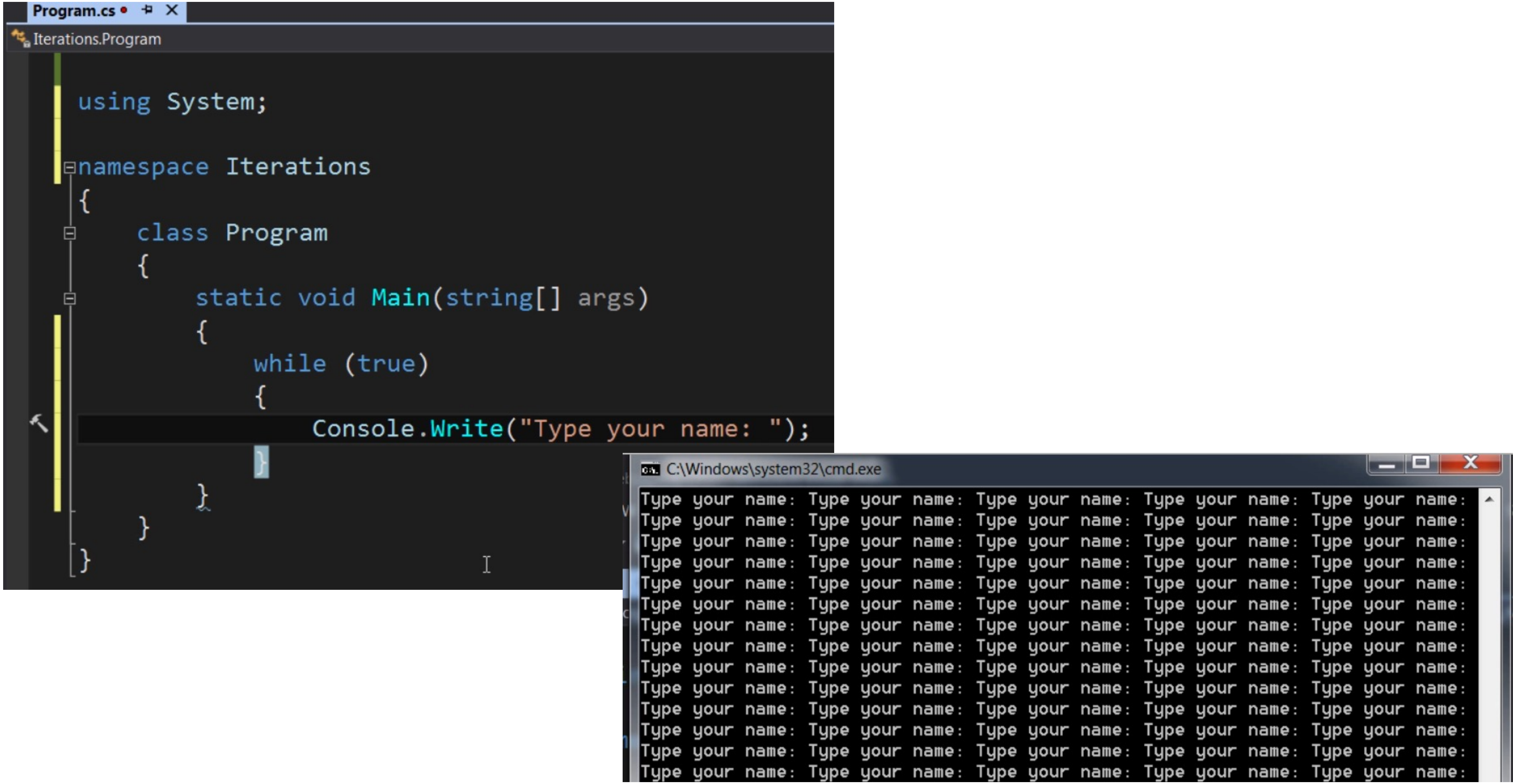
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            var numbers = new int[] {1, 2, 3, 4};

            foreach (var number in numbers)
            {
                Console.WriteLine(number);
            }
        }
    }
}
```



```
1
2
3
4
Press any key to continue . . . _
```

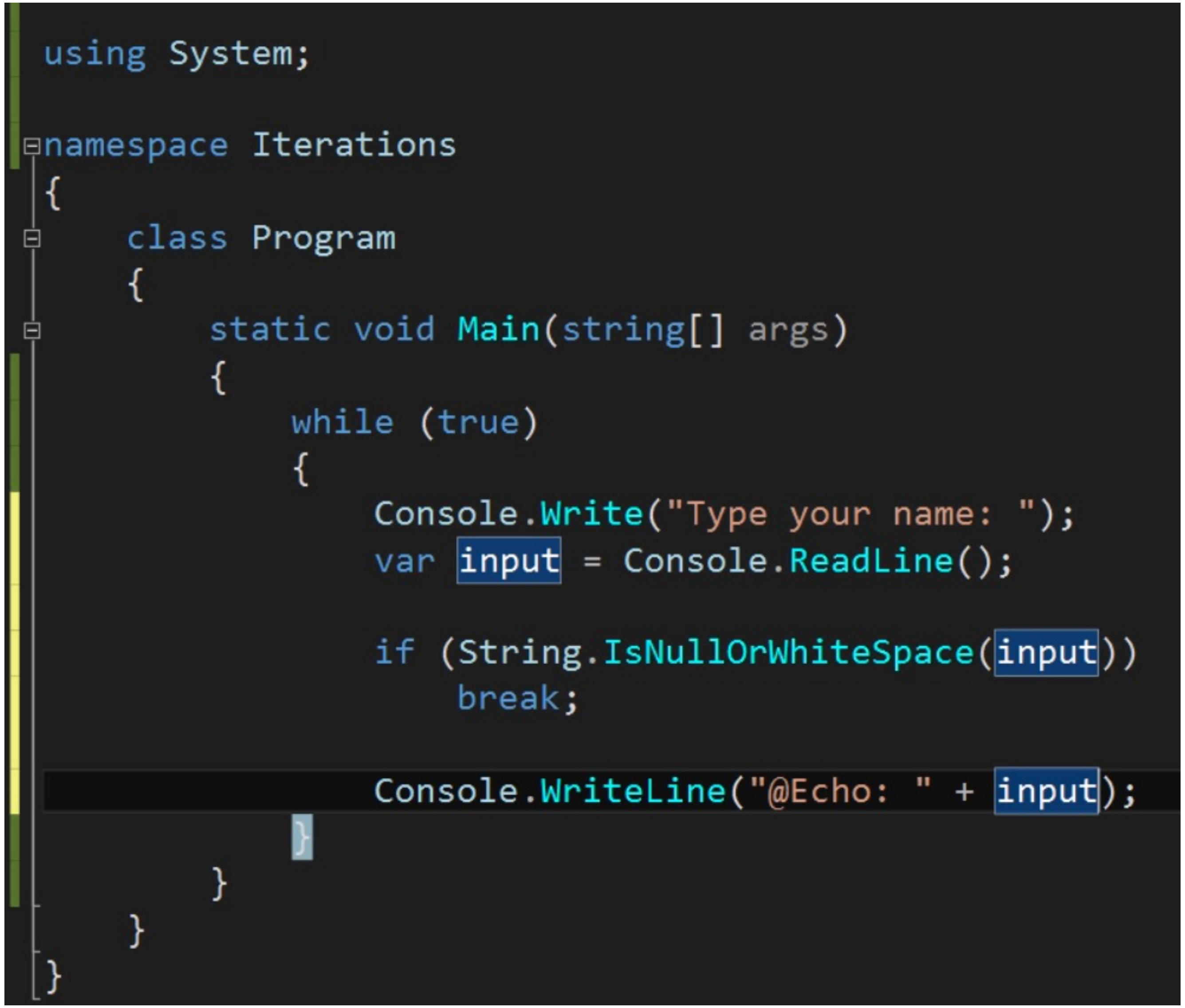


```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            //for (var i = 1; i <= 10; i++)
            //{
            //    if (i% 2 == 0)
            //        Console.WriteLine(i);
            //}

            var i = 0;
            while (i <= 10)
            {
                if (i % 2 == 0)
                    Console.WriteLine(i);

                i++;
            }
        }
    }
}
```

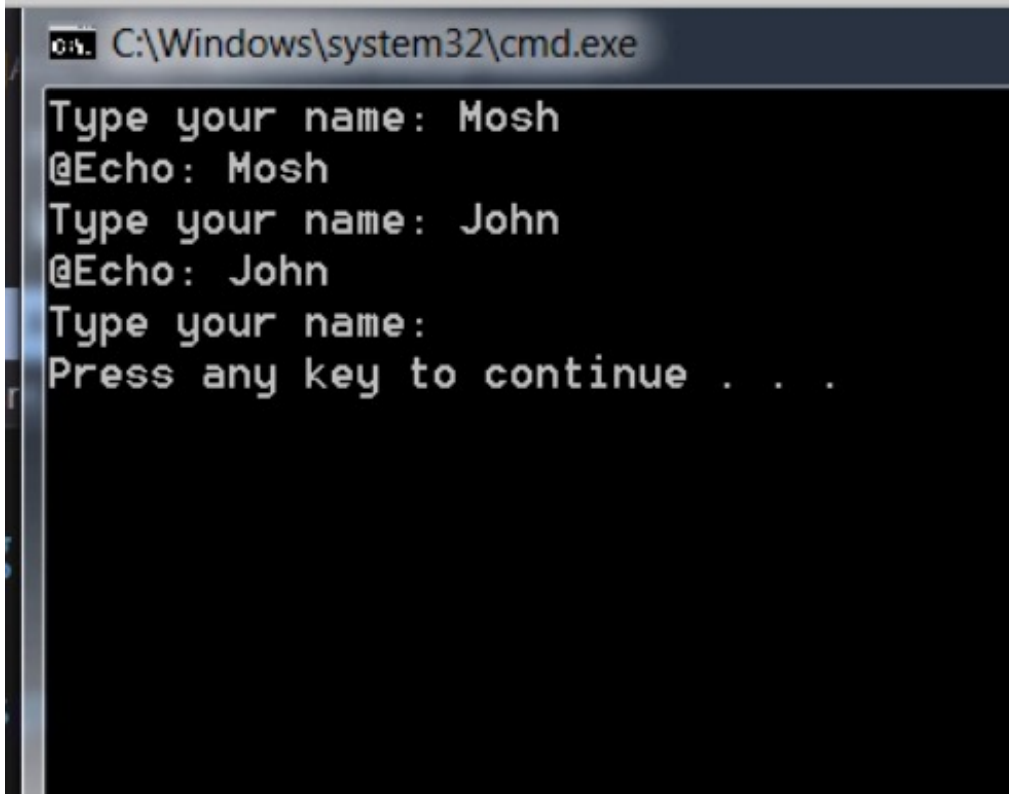


```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                Console.Write("Type your name: ");
            }
        }
    }
}
```

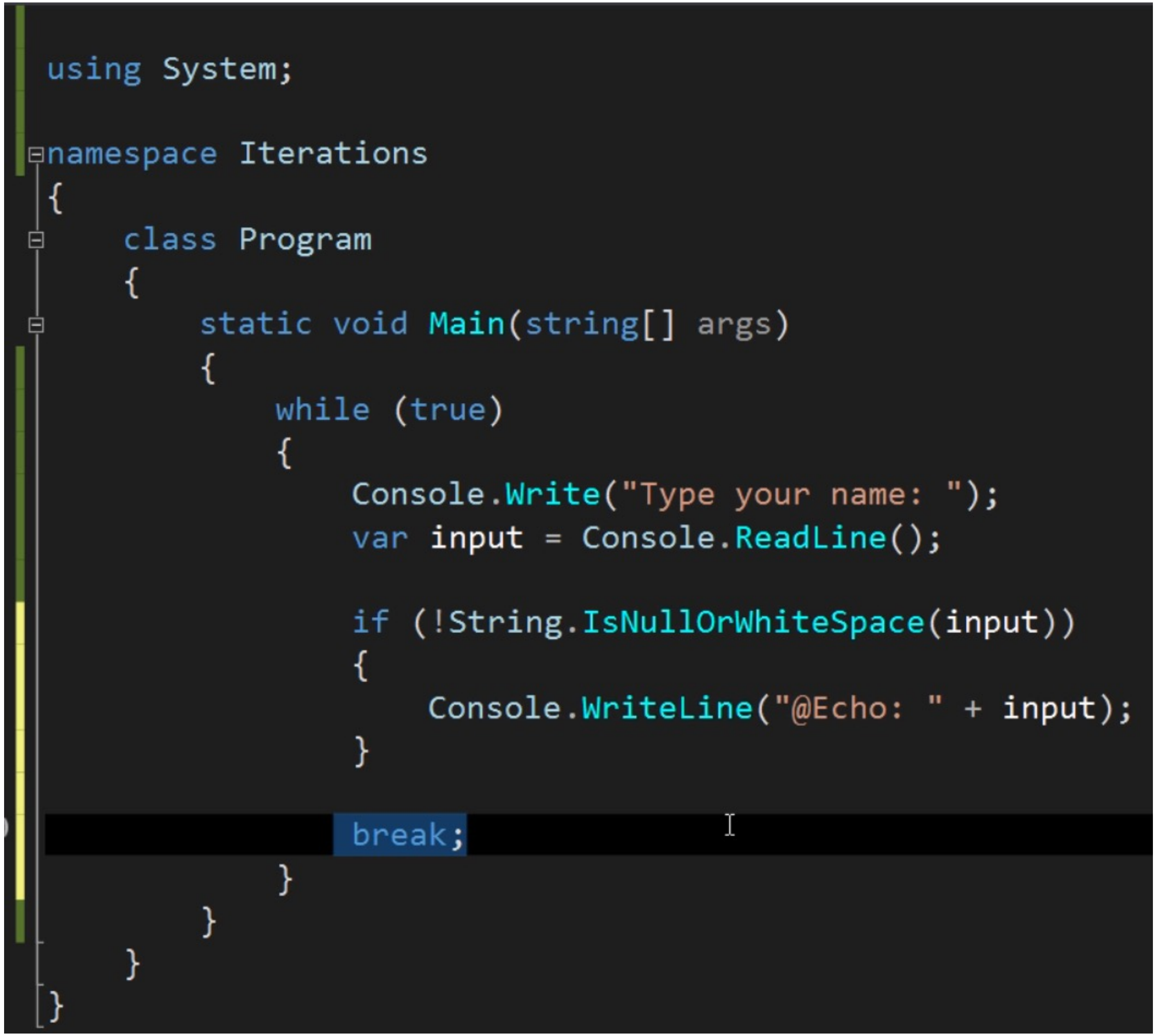
```
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
Type your name: Type your name: Type your name: Type your name: Type your name:
```



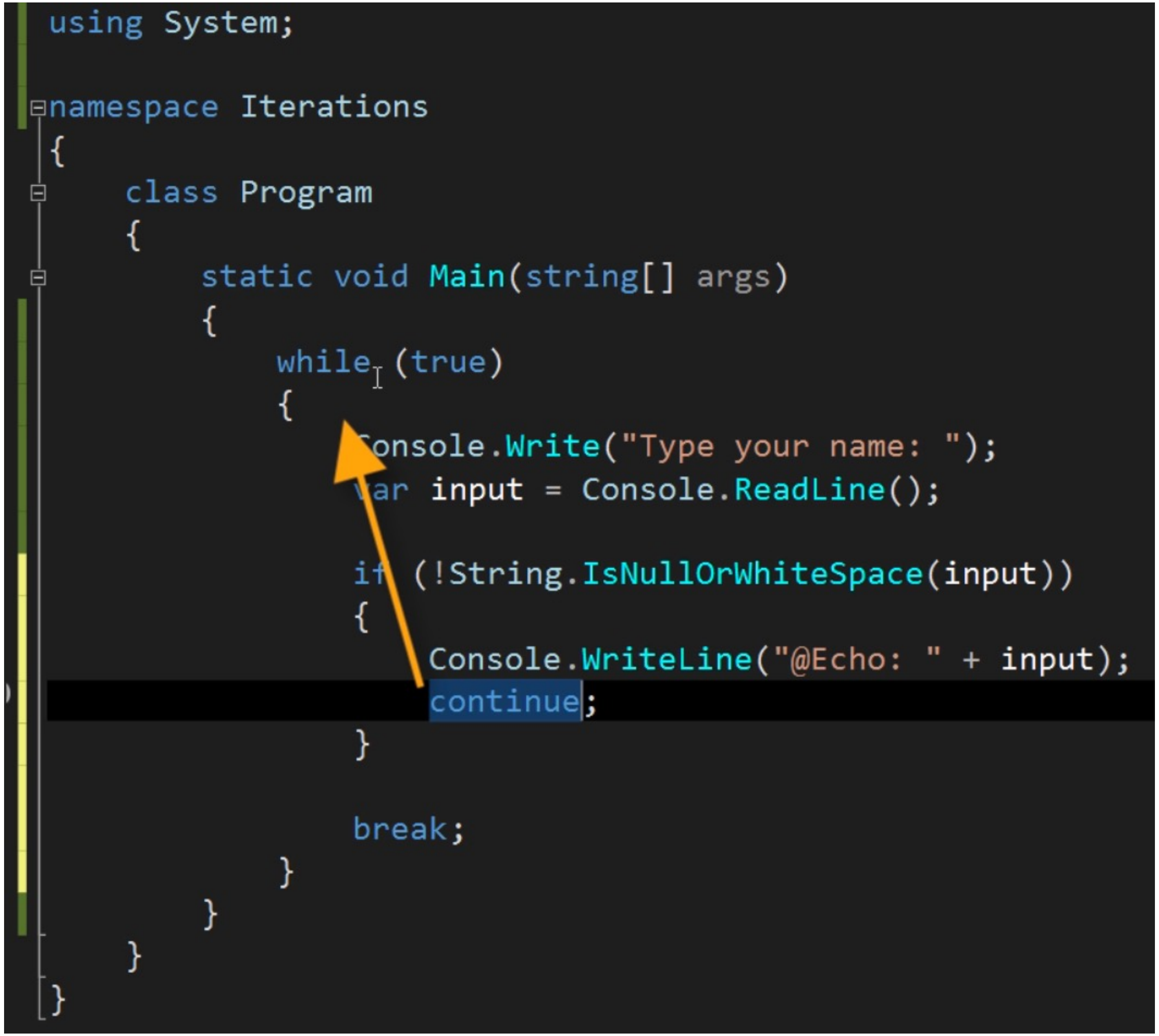
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                Console.Write("Type your name: ");
                var input = Console.ReadLine();

                if (String.IsNullOrWhiteSpace(input))
                    break;

                Console.WriteLine("@Echo: " + input);
            }
        }
    }
}
```



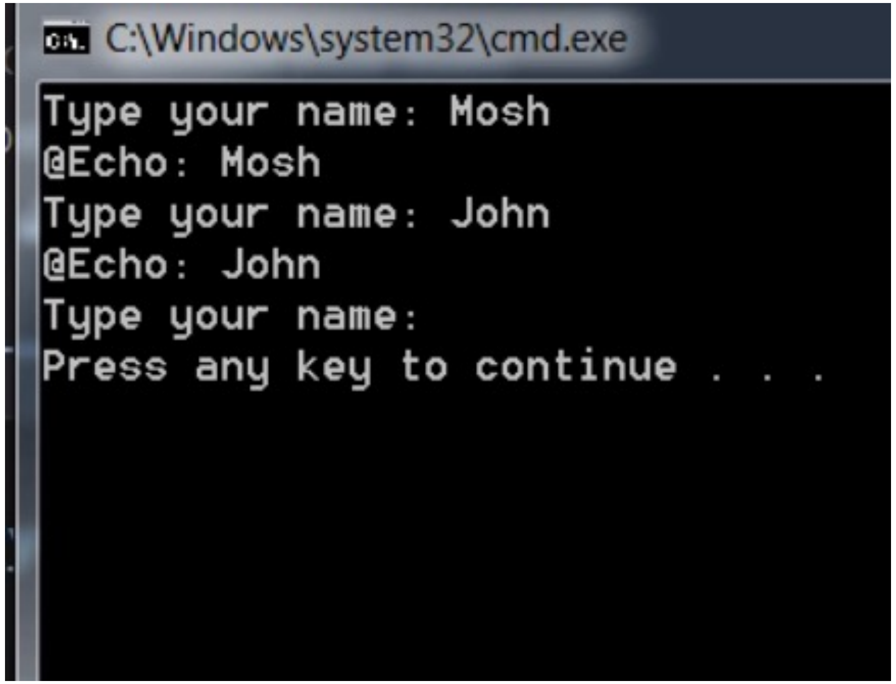
```
Type your name: Mosh
@Echo: Mosh
Type your name: John
@Echo: John
Type your name:
Press any key to continue . . .
```



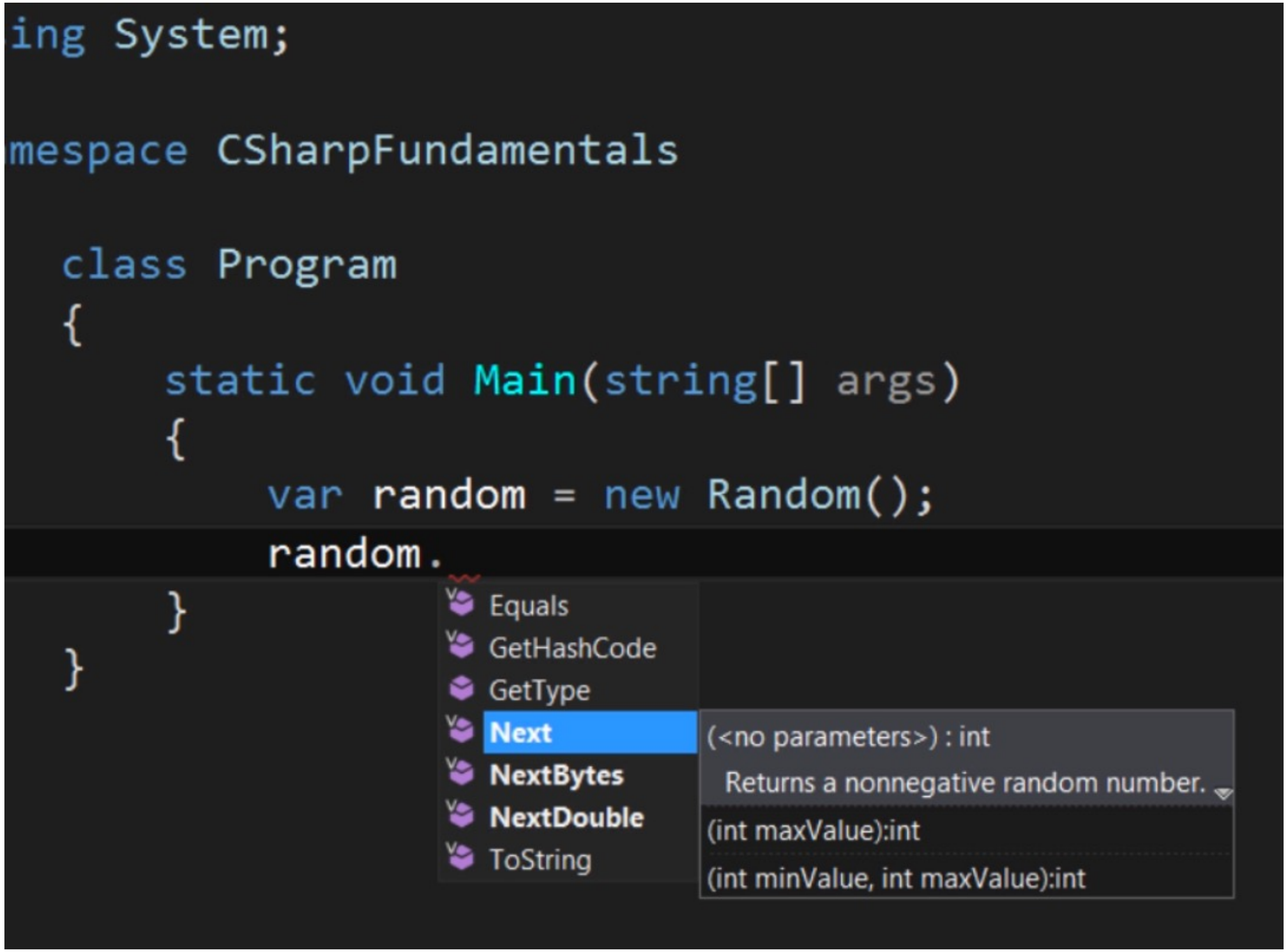
```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                Console.Write("Type your name: ");
                var input = Console.ReadLine();

                if (!String.IsNullOrWhiteSpace(input))
                {
                    Console.WriteLine("@Echo: " + input);
                }

                break;
            }
        }
    }
}
```



```
using System;

namespace Iterations
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                Console.Write("Type your name: ");
                var input = Console.ReadLine();

                if (!String.IsNullOrWhiteSpace(input))
                {
                    Console.WriteLine("@Echo: " + input);
                    continue;
                }

                break;
            }
        }
    }
}
```



```
Type your name: Mosh
@Echo: Mosh
Type your name: John
@Echo: John
Type your name:
Press any key to continue . . .
```



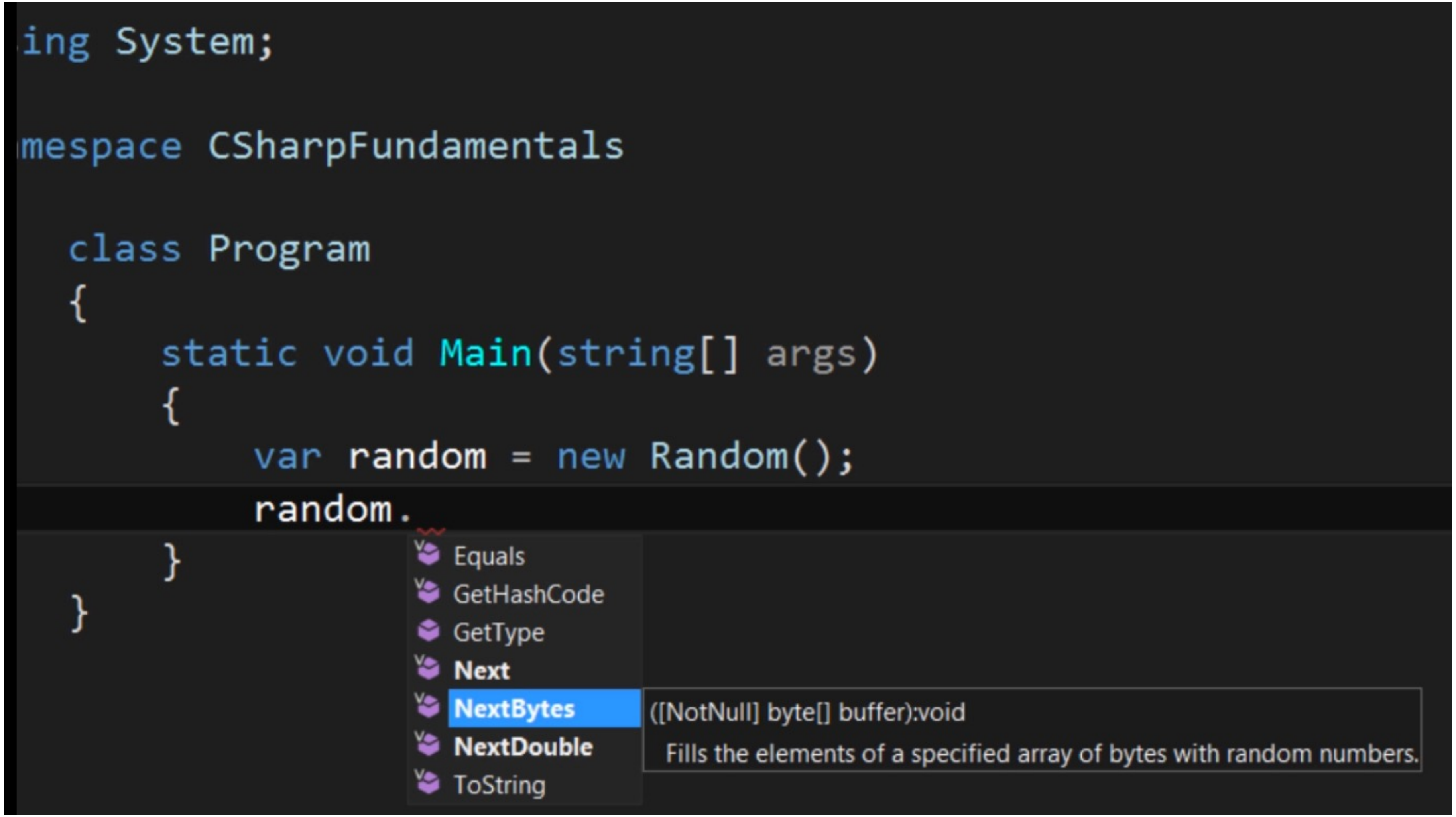
```
ing System;

mespace CSharpFundamentals

  class Program
  {
      static void Main(string[] args)
      {
          var random = new Random();
          random.
      }
  }
```

(<no parameters>) : int

Returns a nonnegative random number.

(int maxValue):int

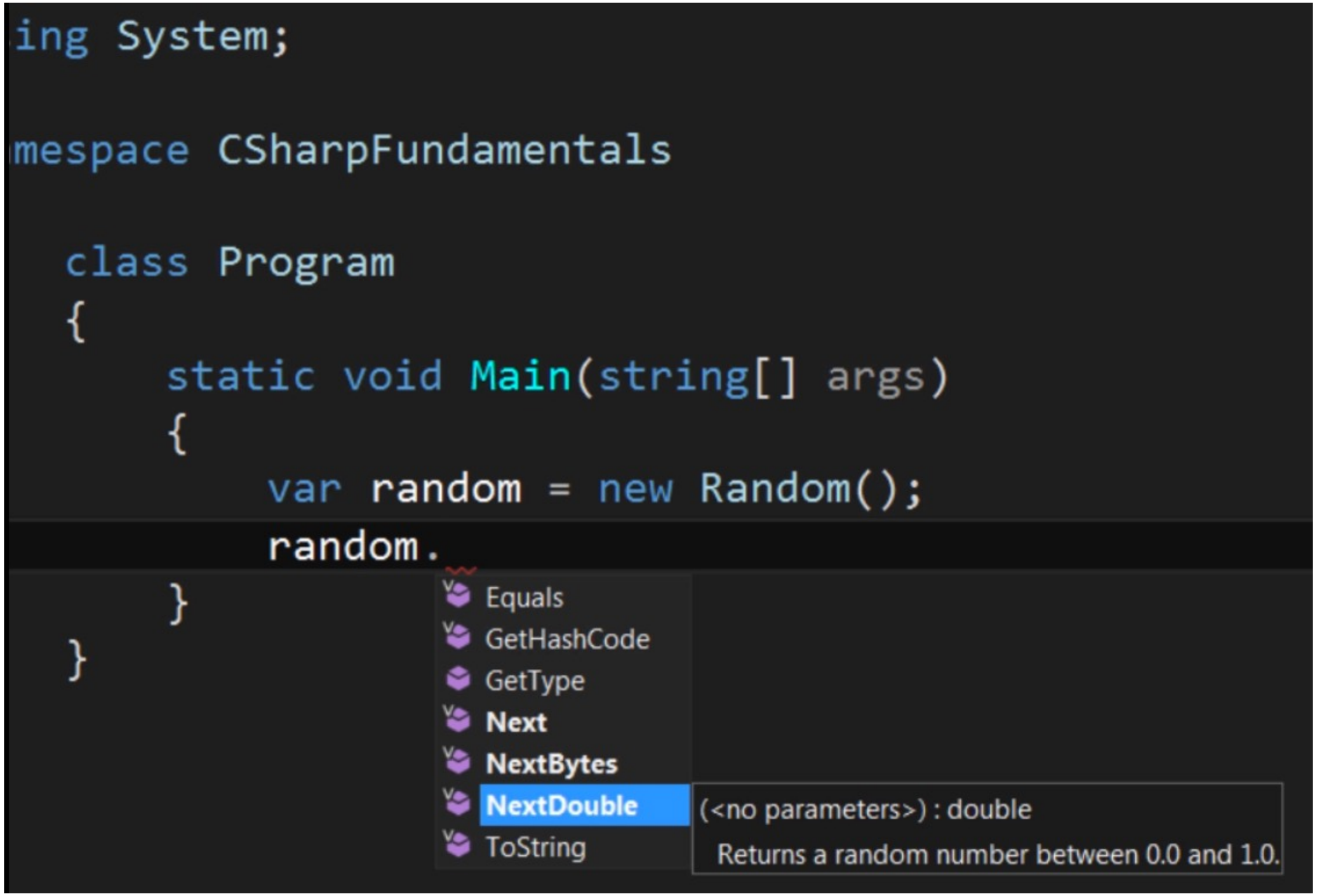
(int minValue, int maxValue):int



```
ing System;

mespace CSharpFundamentals

  class Program
  {
      static void Main(string[] args)
      {
          var random = new Random();
          random.
      }
  }
```

([NotNull] byte[] buffer):void

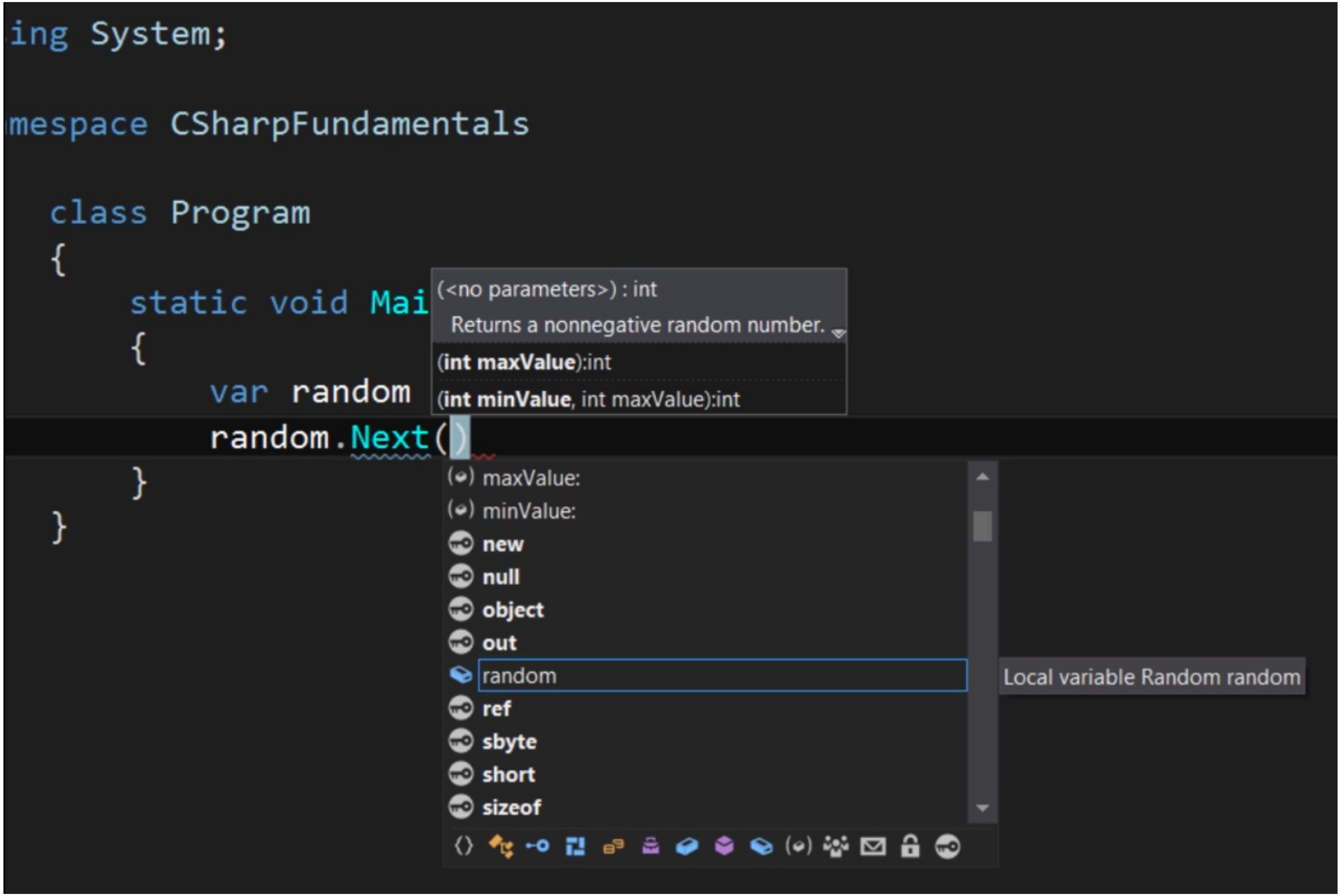
Fills the elements of a specified array of bytes with random numbers.



```
ing System;

mespace CSharpFundamentals

  class Program
  {
      static void Main(string[] args)
      {
          var random = new Random();
          random.
      }
  }
```

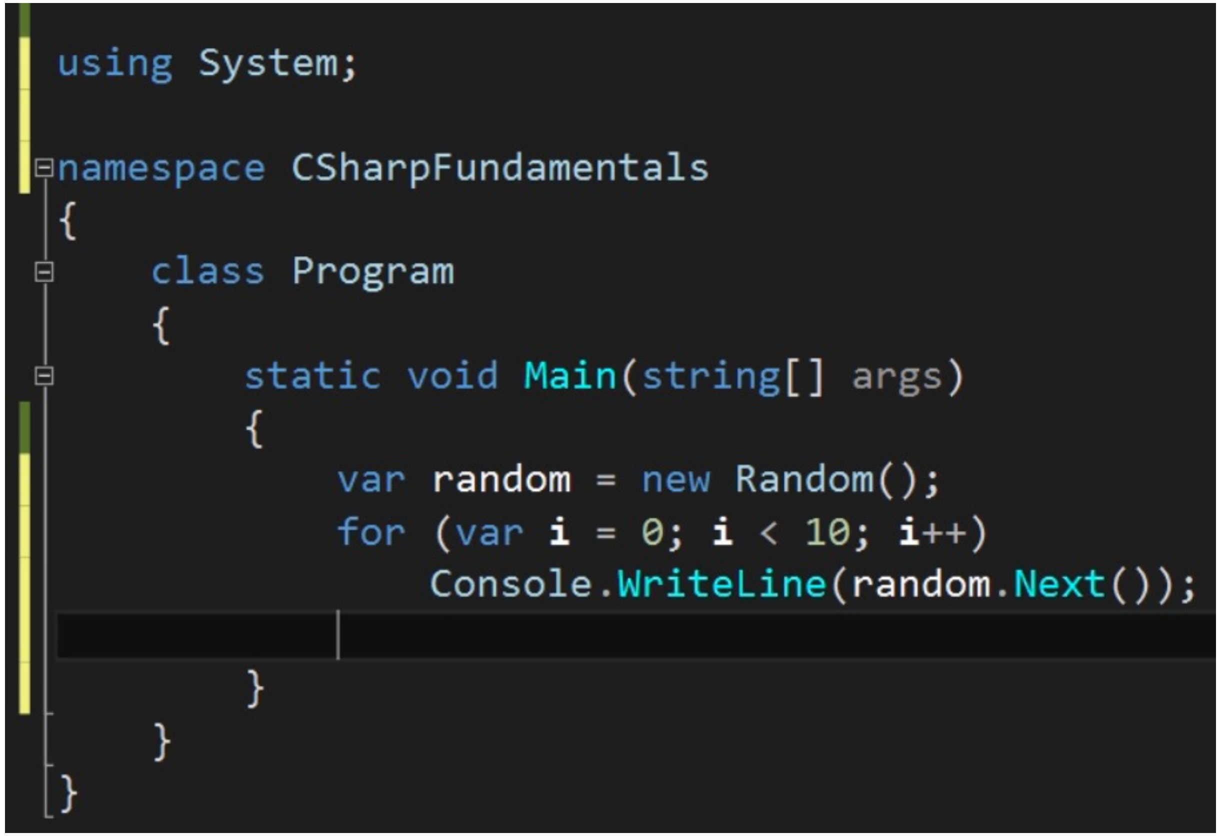
(<no parameters>) : double

Returns a random number between 0.0 and 1.0.

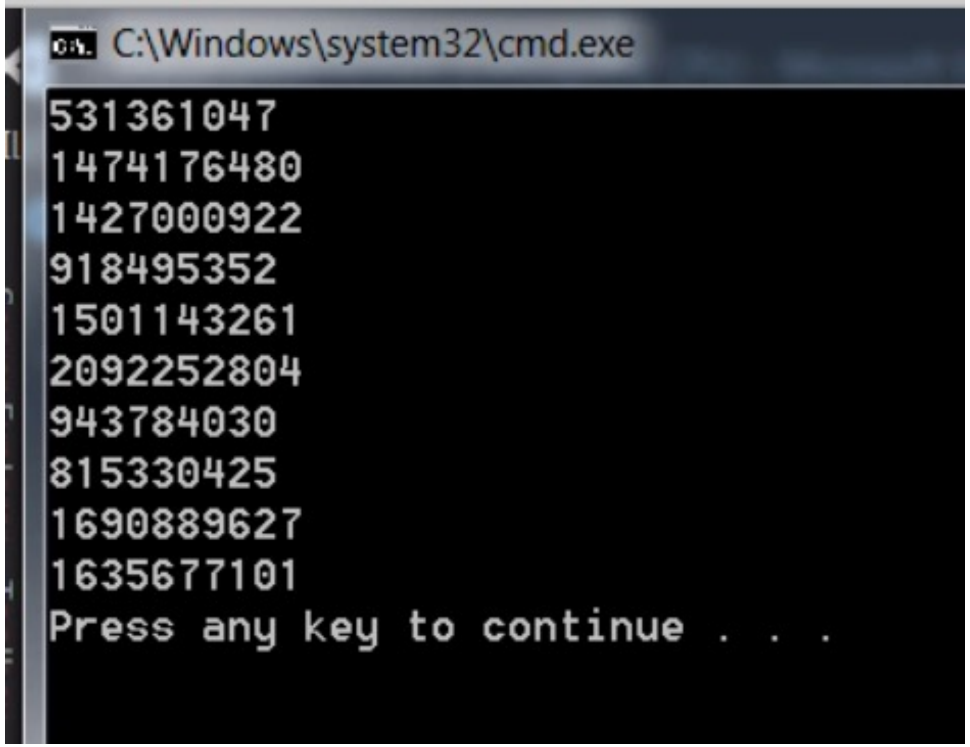


```
ing System;

mespace CSharpFundamentals

  class Program
  {
      static void Mai
      {
          var random
          random.Next()
      }
  }
```

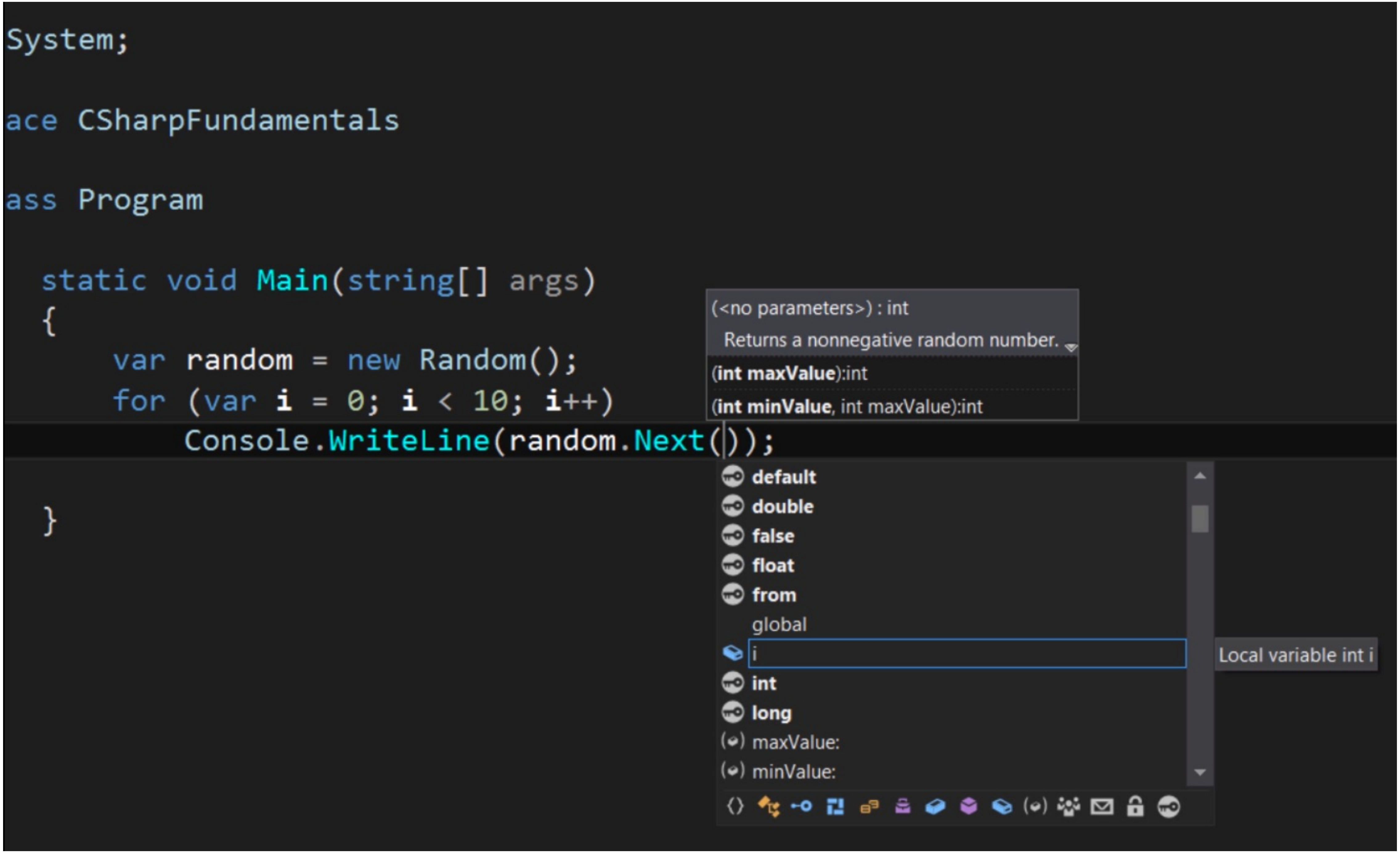
(<no parameters>) : int
Returns a nonnegative random number.
(**int maxValue**):int
(**int minValue**, int maxValue):int



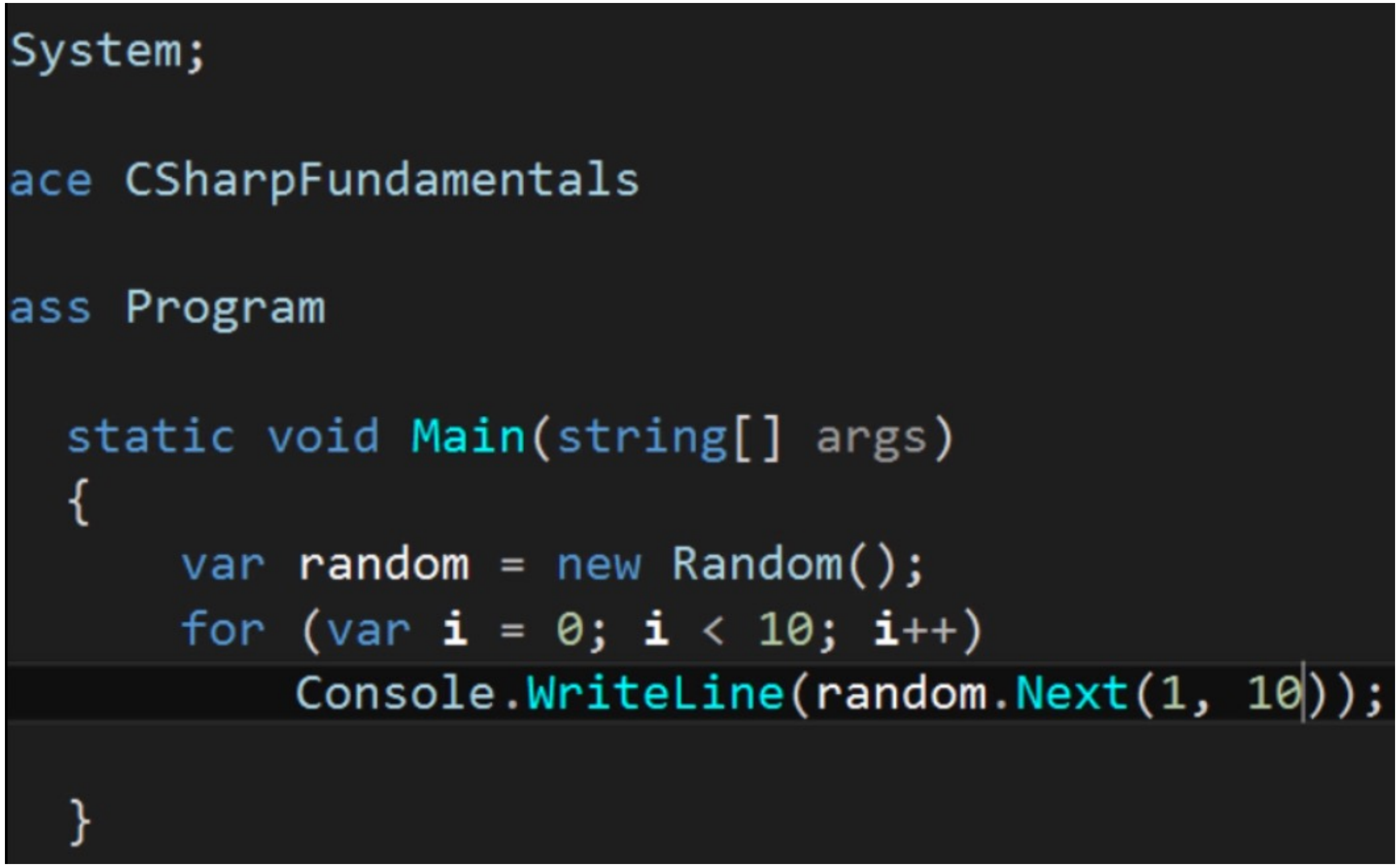
```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();
            for (var i = 0; i < 10; i++)
                Console.WriteLine(random.Next());

        }
    }
}
```



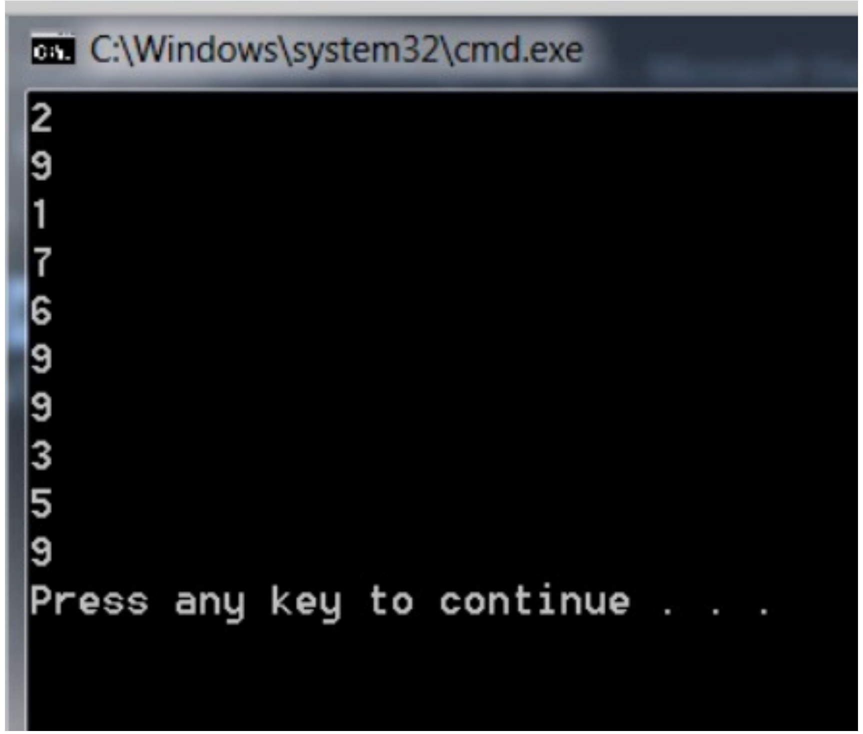
```
531361047
1474176480
1427000922
918495352
1501143261
2092252804
943784030
815330425
1690889627
1635677101
Press any key to continue . . .
```



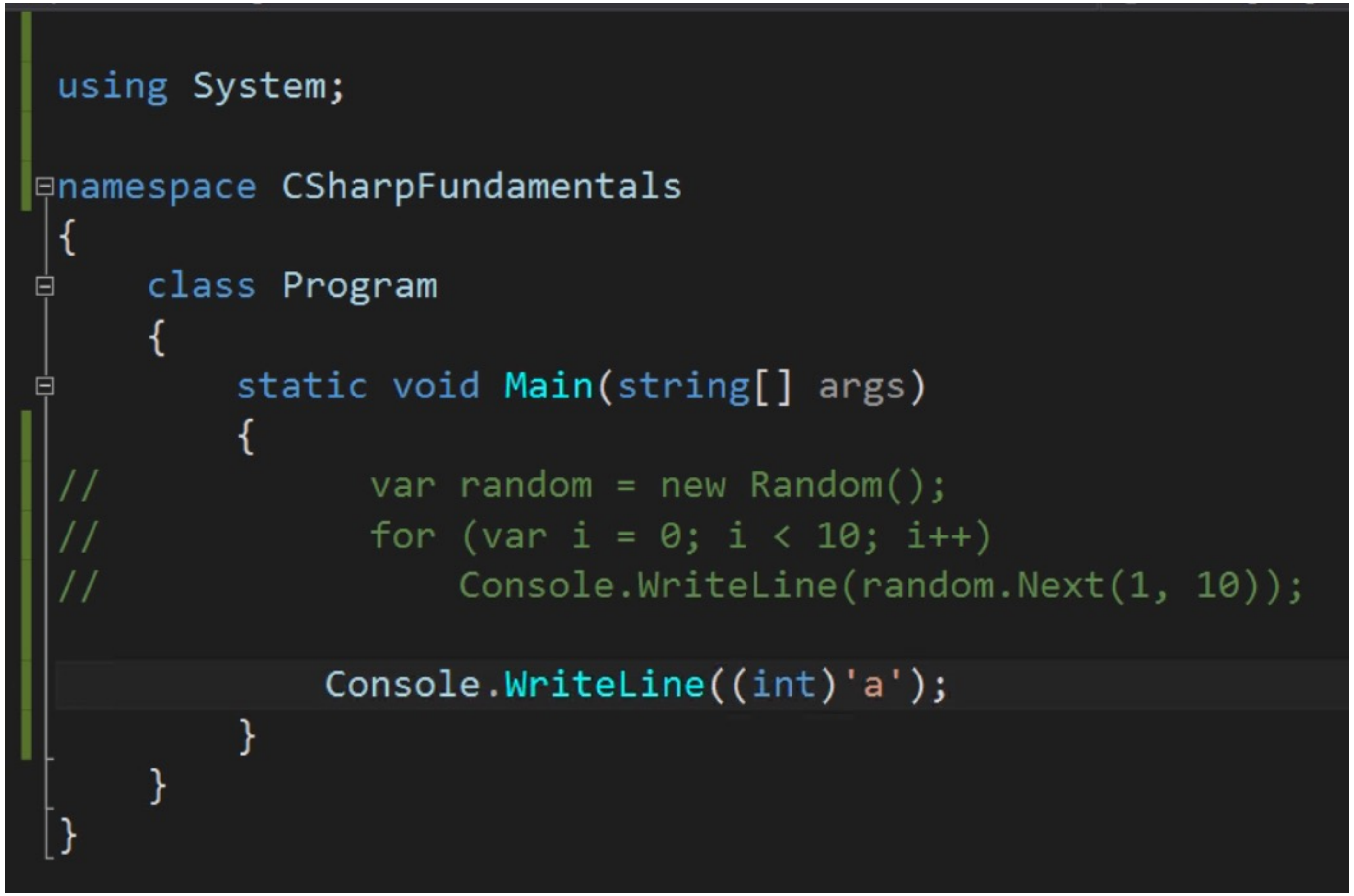
```
System;

ace CSharpFundamentals

ass Program

  static void Main(string[] args)
  {
      var random = new Random();
      for (var i = 0; i < 10; i++)
          Console.WriteLine(random.Next());

  }
```



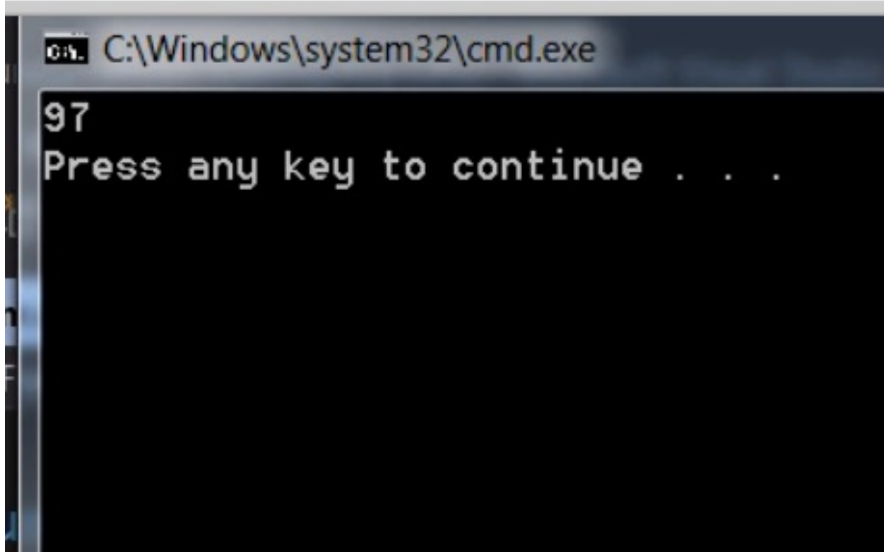
```
System;

ace CSharpFundamentals

ass Program

  static void Main(string[] args)
  {
      var random = new Random();
      for (var i = 0; i < 10; i++)
          Console.WriteLine(random.Next(1, 10));

  }
```



```
2
9
1
7
6
9
9
3
5
9
Press any key to continue . . .
```

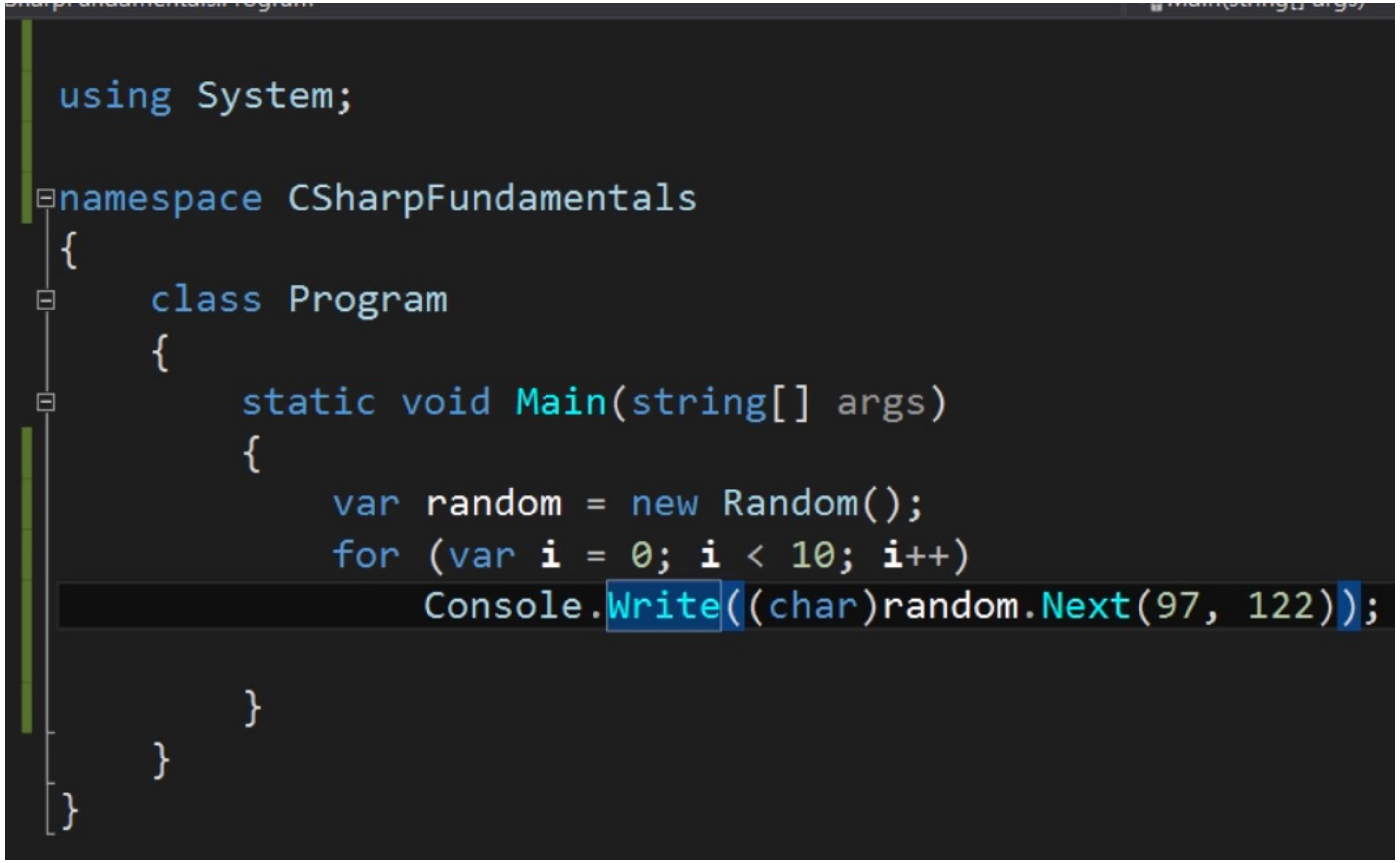


```csharp
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
//            var random = new Random();
//            for (var i = 0; i < 10; i++)
//                Console.WriteLine(random.Next(1, 10));

            Console.WriteLine((int)'a');
        }
    }
}
```

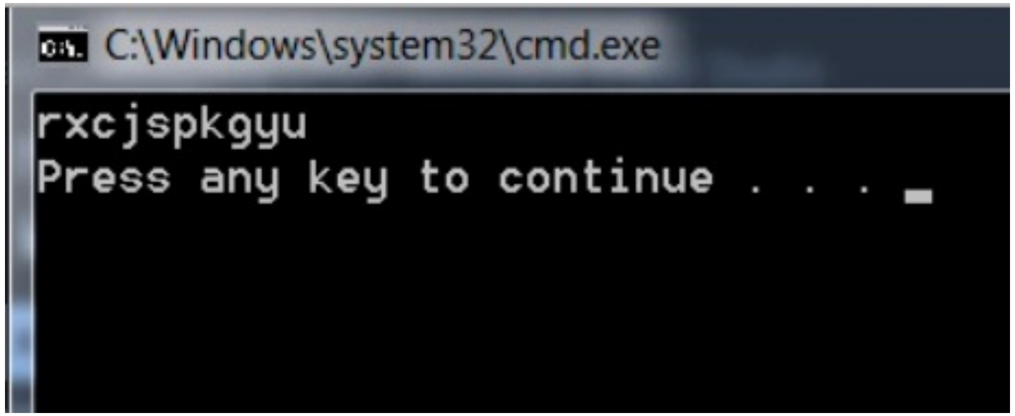


C:\Windows\system32\cmd.exe

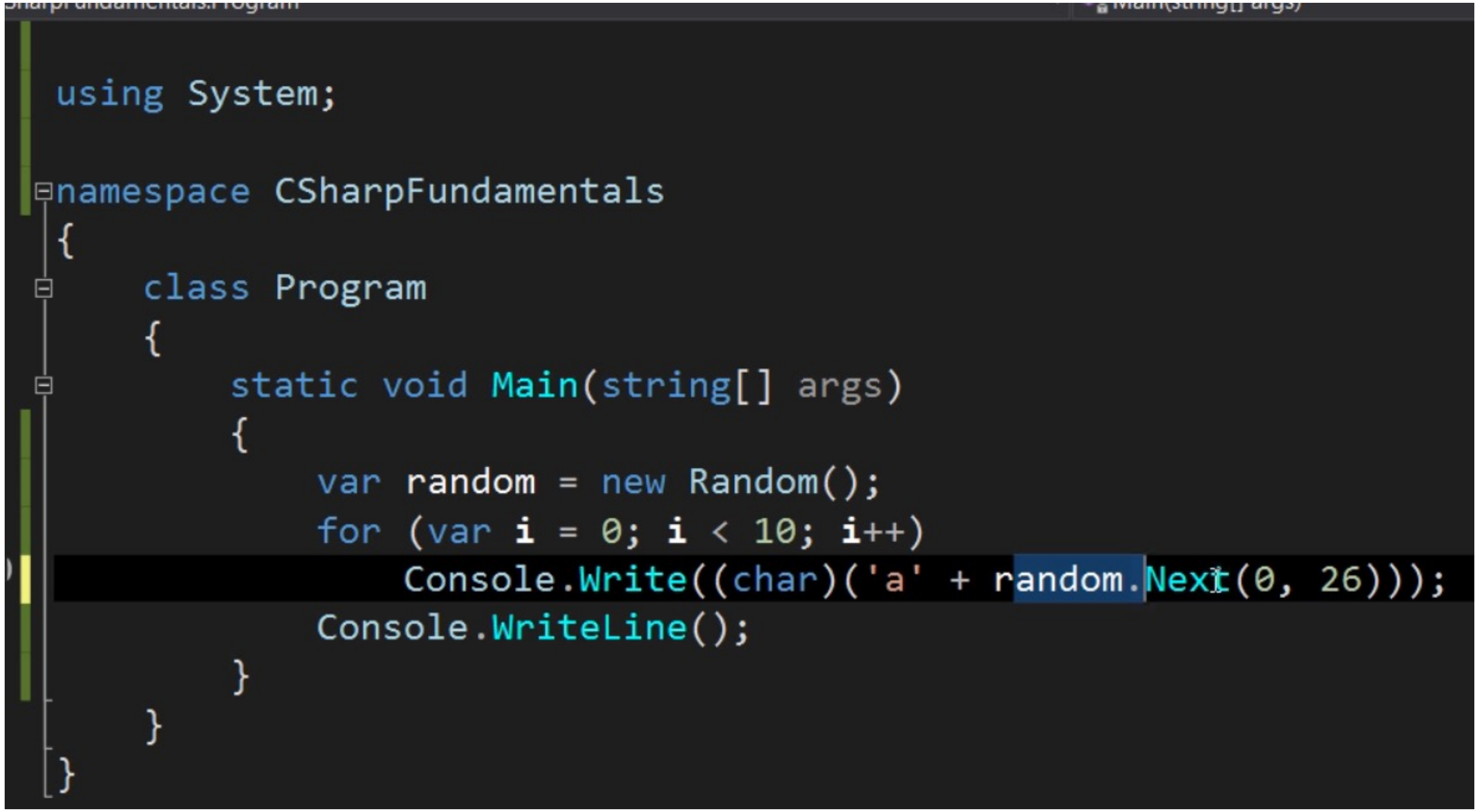
```
97
Press any key to continue . . .
```



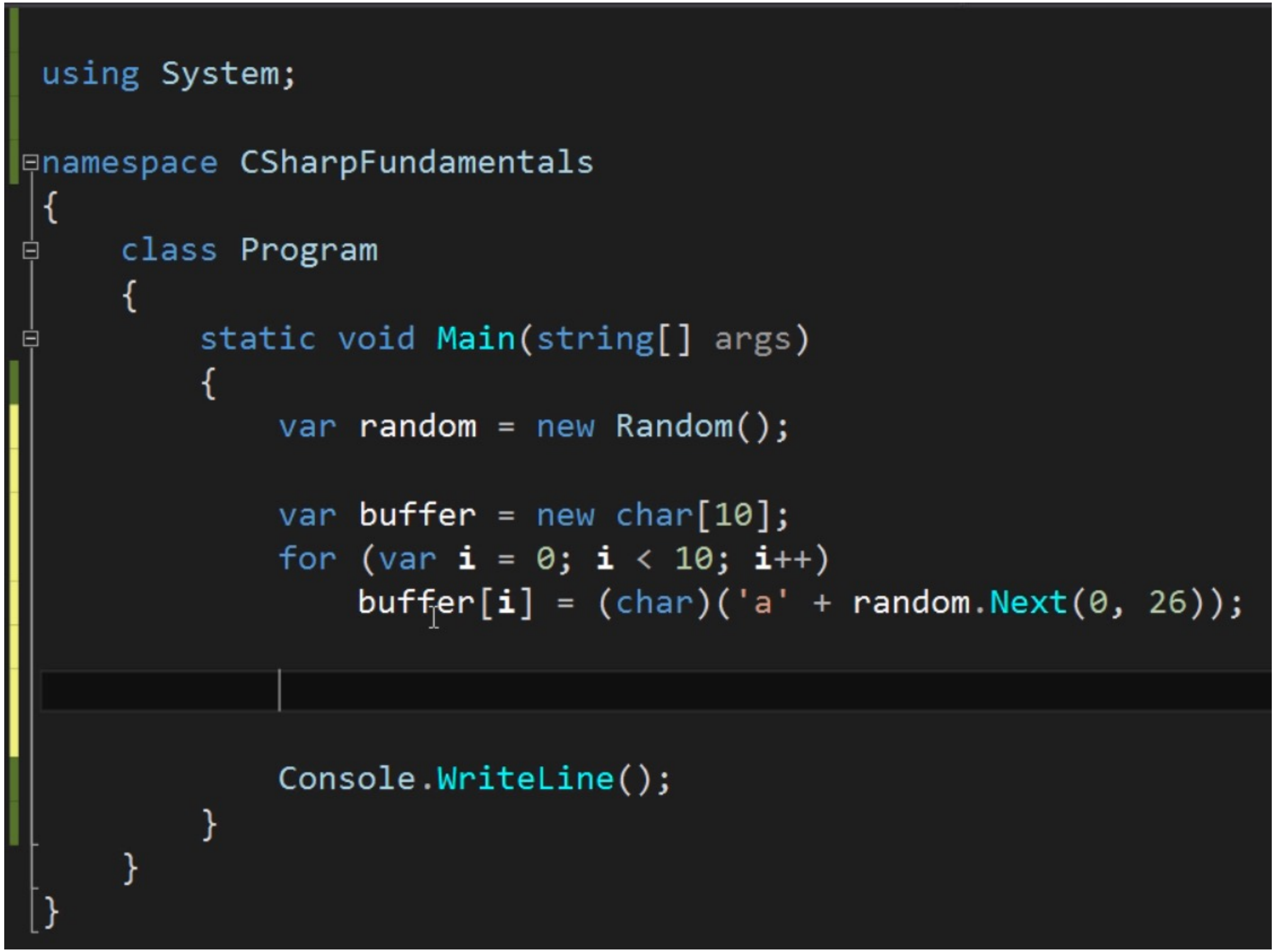
```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();
            for (var i = 0; i < 10; i++)
                Console.Write((char)random.Next(97, 122));

        }
    }
}
```



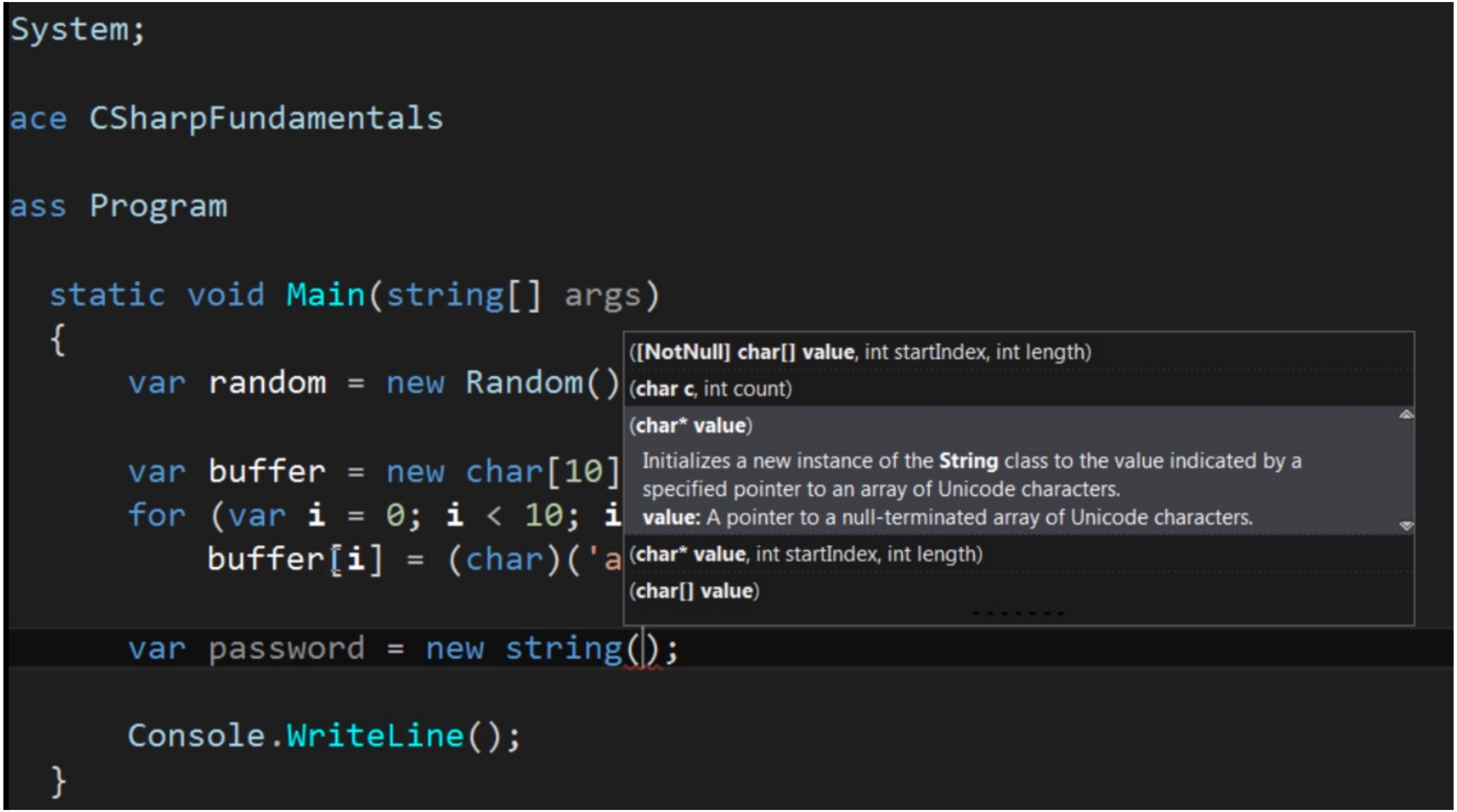
```
rxcjspkgyu
Press any key to continue . . .
```



```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();
            for (var i = 0; i < 10; i++)
                Console.Write((char)('a' + random.Next(0, 26)));
            Console.WriteLine();
        }
    }
}
```



```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();

            var buffer = new char[10];
            for (var i = 0; i < 10; i++)
                buffer[i] = (char)('a' + random.Next(0, 26));



            Console.WriteLine();
        }
    }
}
```



```
System;

ace CSharpFundamentals

ass Program

  static void Main(string[] args)
  {
      var random = new Random()

      var buffer = new char[10]
      for (var i = 0; i < 10; i
          buffer[i] = (char)('a

      var password = new string();

      Console.WriteLine();
  }
```

([NotNull] char[] value, int startIndex, int length)

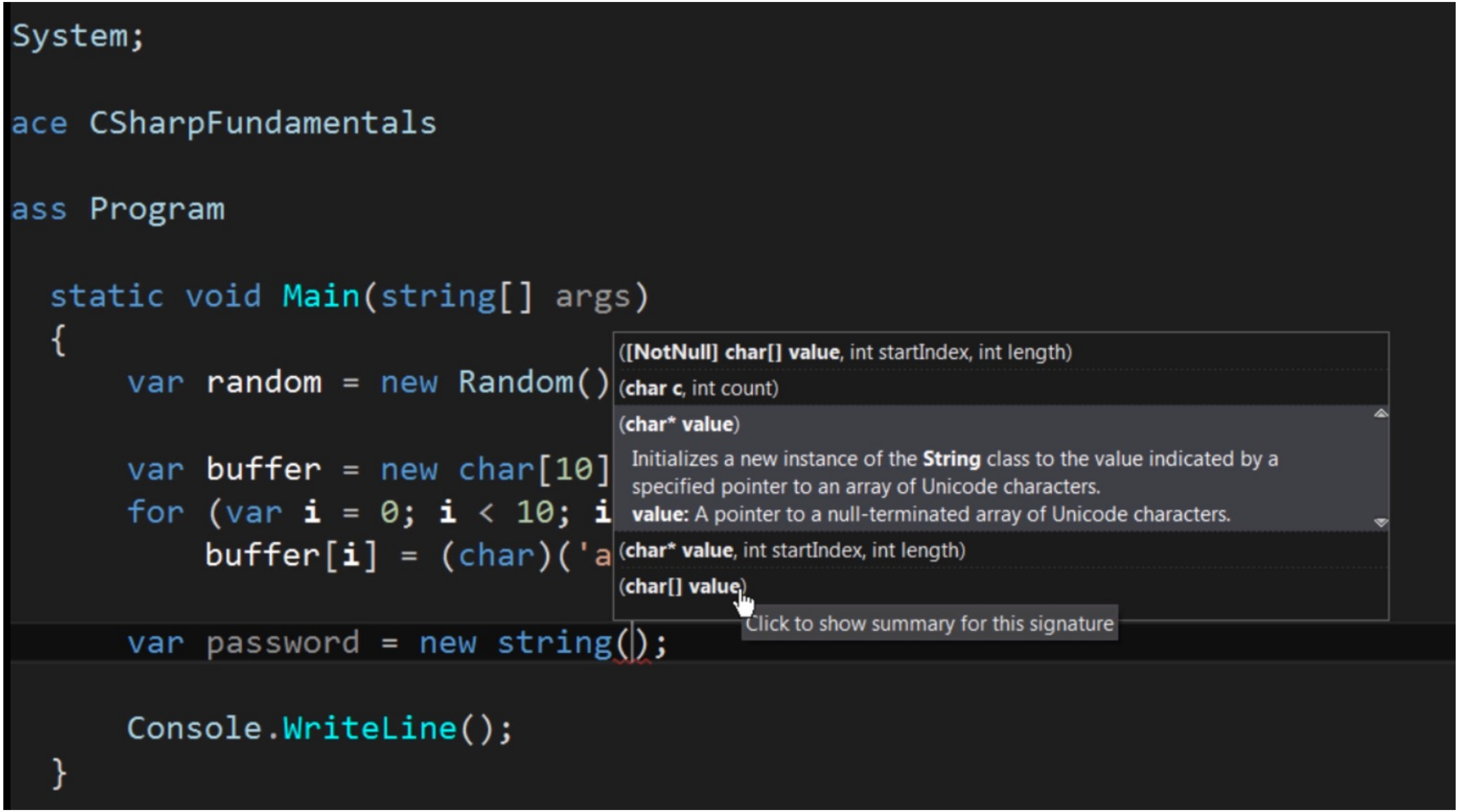
(char c, int count)

(char* value)

Initializes a new instance of the **String** class to the value indicated by a specified pointer to an array of Unicode characters.
value: A pointer to a null-terminated array of Unicode characters.

(char* value, int startIndex, int length)

(char[] value)



```
System;

ace CSharpFundamentals

ass Program

 static void Main(string[] args)
 {
   var random = new Random()

   var buffer = new char[10]
   for (var i = 0; i < 10; i
     buffer[i] = (char)('a

   var password = new string();

   Console.WriteLine();
 }
```

([NotNull] char[] value, int startIndex, int length)

(char c, int count)

(char* value)

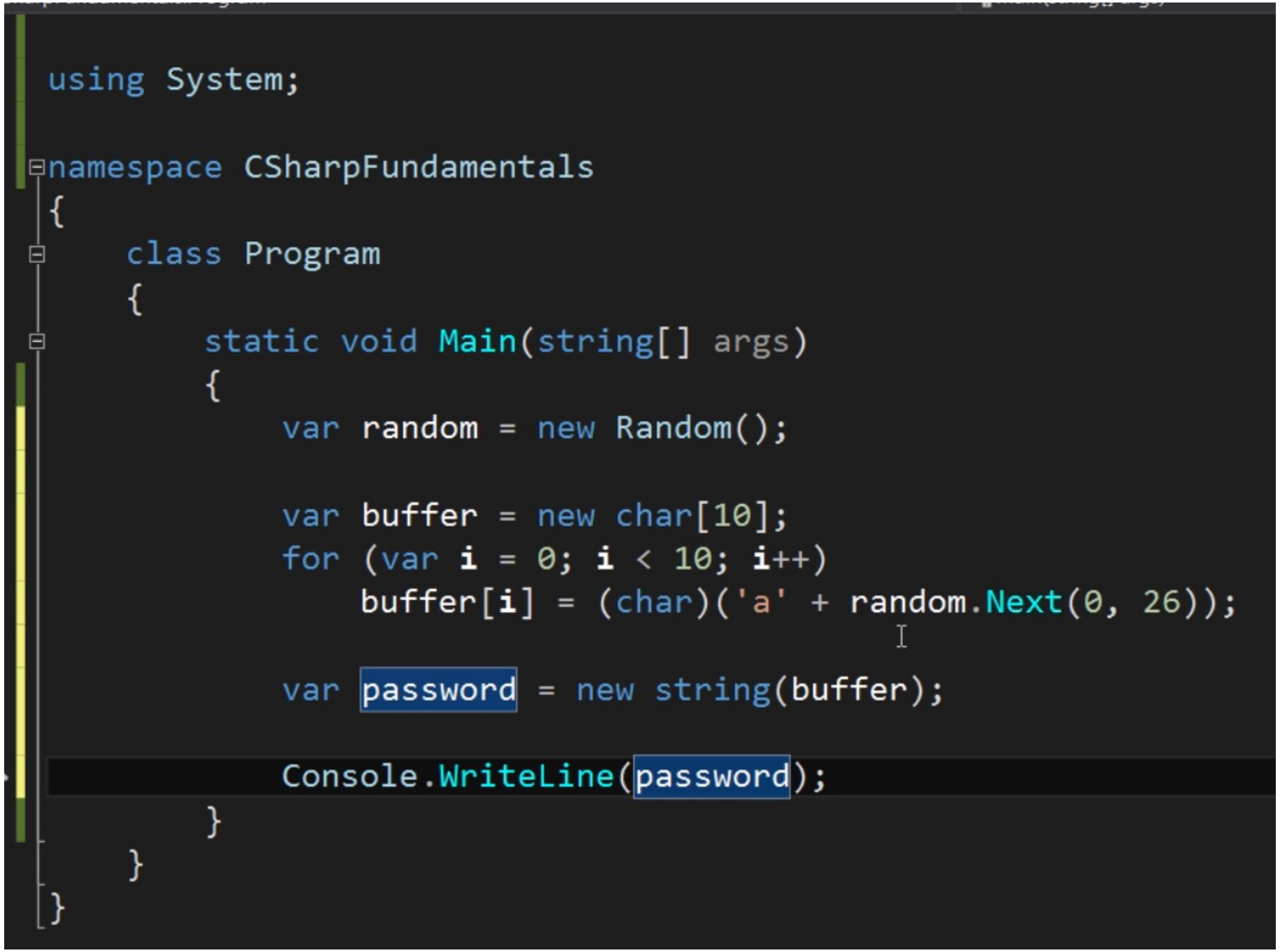
Initializes a new instance of the **String** class to the value indicated by a specified pointer to an array of Unicode characters.

value: A pointer to a null-terminated array of Unicode characters.

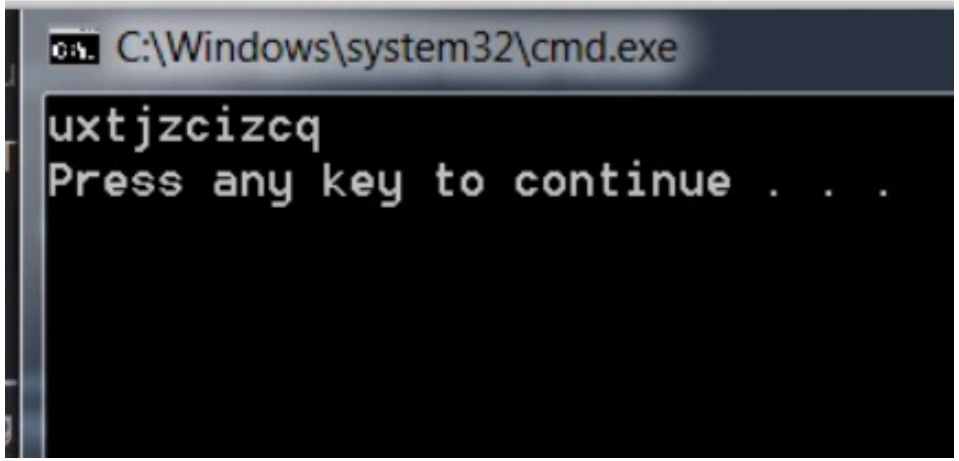
(char* value, int startIndex, int length)

(char[] value)

Click to show summary for this signature

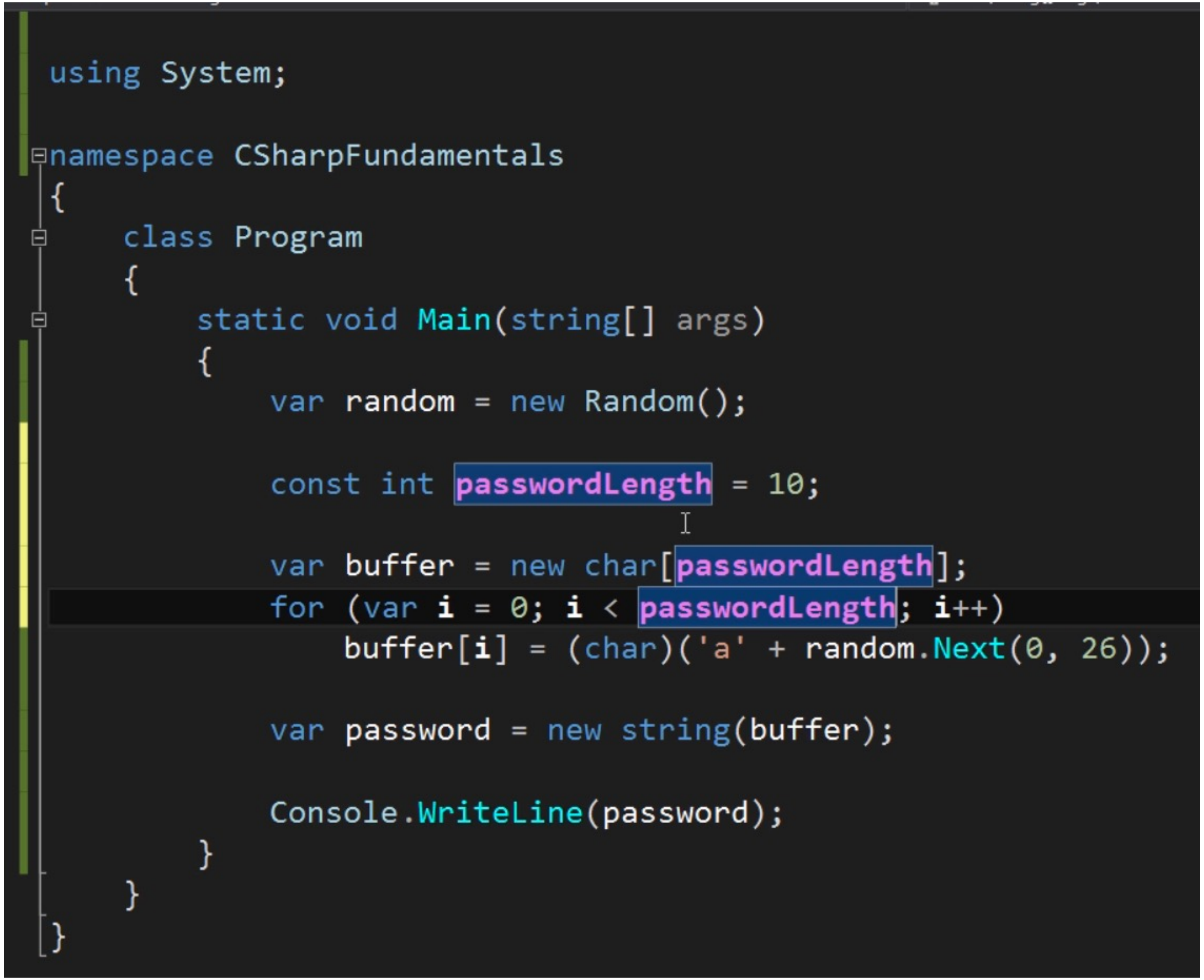


```csharp
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();

            var buffer = new char[10];
            for (var i = 0; i < 10; i++)
                buffer[i] = (char)('a' + random.Next(0, 26));

            var password = new string(buffer);

            Console.WriteLine(password);
        }
    }
}
```



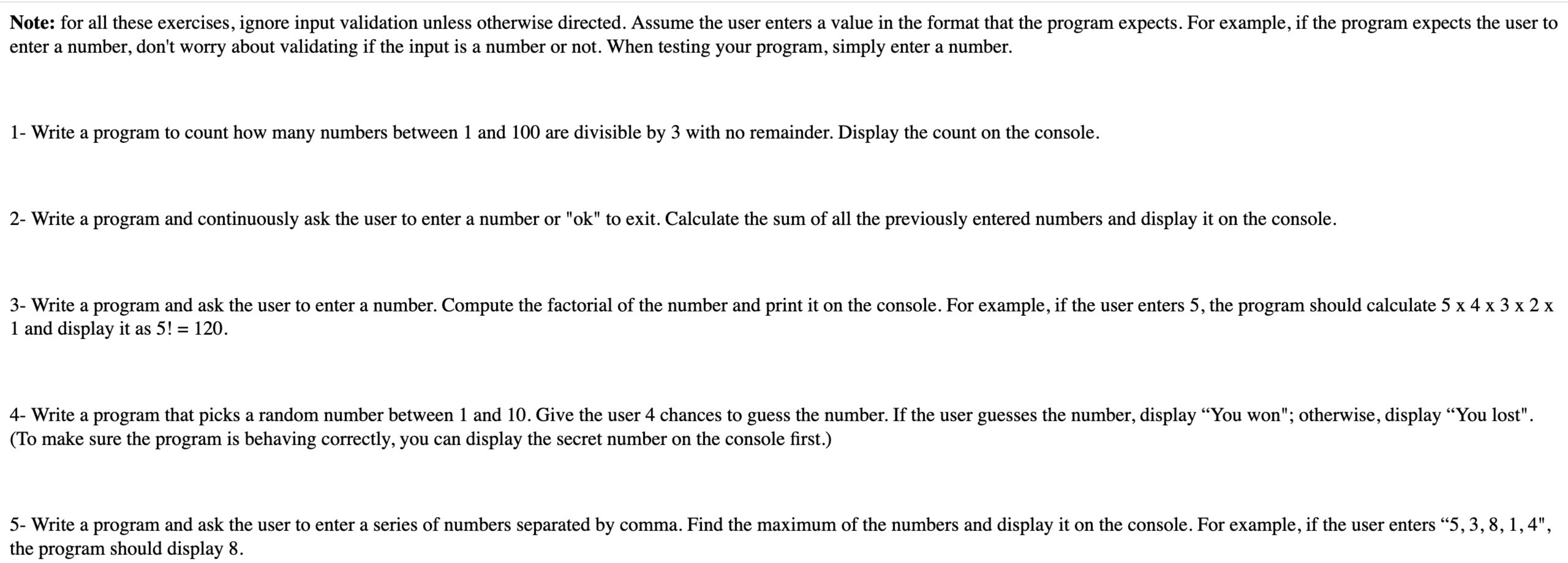
C:\Windows\system32\cmd.exe

```
uxtjzcizcq
Press any key to continue . . .
```



```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var random = new Random();

            const int passwordLength = 10;

            var buffer = new char[passwordLength];
            for (var i = 0; i < passwordLength; i++)
                buffer[i] = (char)('a' + random.Next(0, 26));

            var password = new string(buffer);

            Console.WriteLine(password);
        }
    }
}
```

Excersise



Note: for all these exercises, ignore input validation unless otherwise directed. Assume the user enters a value in the format that the program expects. For example, if the program expects the user to enter a number, don't worry about validating if the input is a number or not. When testing your program, simply enter a number.

1- Write a program to count how many numbers between 1 and 100 are divisible by 3 with no remainder. Display the count on the console.

2- Write a program and continuously ask the user to enter a number or "ok" to exit. Calculate the sum of all the previously entered numbers and display it on the console.

3- Write a program and ask the user to enter a number. Compute the factorial of the number and print it on the console. For example, if the user enters 5, the program should calculate 5 x 4 x 3 x 2 x 1 and display it as 5! = 120.

4- Write a program that picks a random number between 1 and 10. Give the user 4 chances to guess the number. If the user guesses the number, display “You won"; otherwise, display “You lost". (To make sure the program is behaving correctly, you can display the secret number on the console first.)

5- Write a program and ask the user to enter a series of numbers separated by comma. Find the maximum of the numbers and display it on the console. For example, if the user enters “5, 3, 8, 1, 4", the program should display 8.