C# for Visual Basic .NET Developers

Basic C# Syntax

Craig Shoemaker
craigshoemaker.net
@craigshoemaker

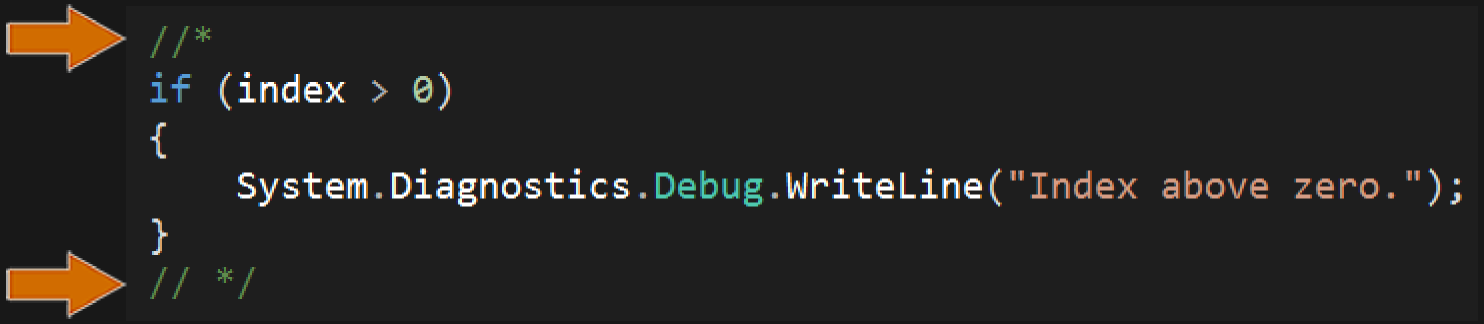
pluralsight
hardcore developer training

```
' Changing the following code will
' change the world. Okay - just your
' world and you can kiss your nights
' and weekends goodbye.
' Changing the following code will
' change the world. Okay - just your
' world and you can kiss your nights
' and weekends goodbye.
```

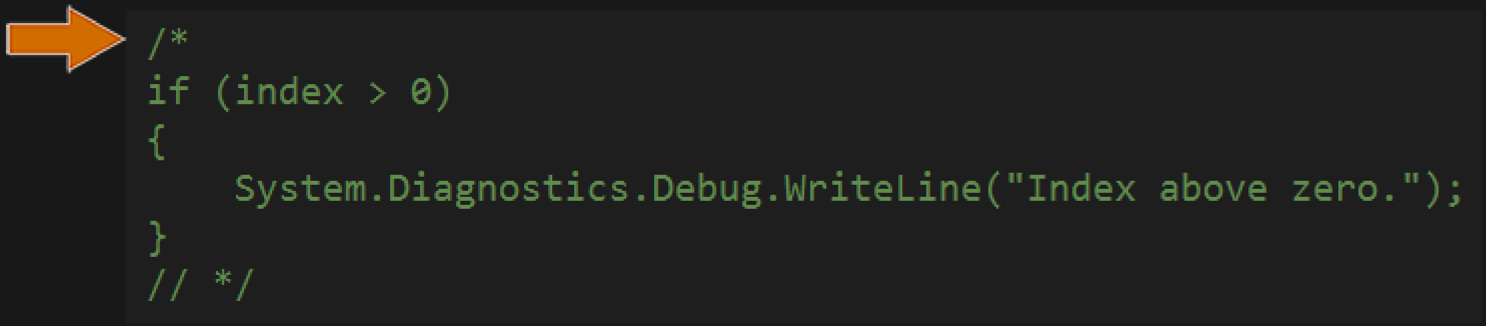
```
// Changing the following code will
// change the world. Okay - just your
// world and you can kiss your nights
// and weekends goodbye.
```

```
' Changing the following code will
' change the world. Okay - just your
' world and you can kiss your nights
' and weekends goodbye.
```

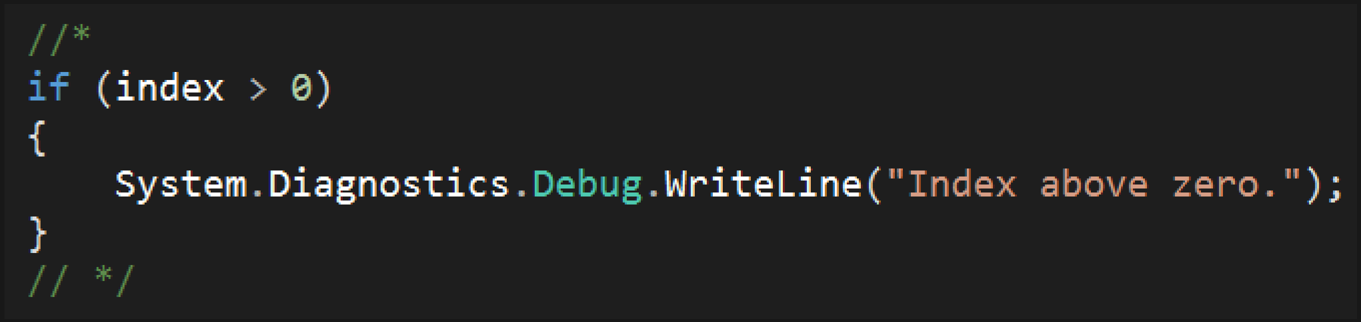
```
/*
 Changing the following code will
 change the world. Okay - just your
 world and you can kiss your nights
 and weekends goodbye.
 */
```



```
//*
if (index > 0)
{
    System.Diagnostics.Debug.WriteLine("Index above zero.");
}
// */
```



```
/*
if (index > 0)
{
    System.Diagnostics.Debug.WriteLine("Index above zero.");
}
// */
```



```
//*
if (index > 0)
{
    System.Diagnostics.Debug.WriteLine("Index above zero.");
}
// */
```

```vb
Private _boolean As Boolean
Private _byte As Byte
Private _sbyte As SByte
Private _char As Char
Private _date As Date
Private _decimal As Decimal
Private _double As Double
Private _integer As Integer
Private _uinteger As UInteger
Private _long As Long
Private _ulong As ULong
Private _short As Short
Private _ushort As UShort
Private _object As Object
Private _single As Single
Private _string As String
```

```csharp
private bool _boolean;
private byte _byte;
private sbyte _sbyte;
private char _char;
private DateTime _date;
private decimal _decimal;
private double _double;
private int _integer;
private uint _uinteger;
private long _long;
private ulong _ulong;
private short _short;
private ushort _ushort;
private object _object;
private float _single;
private string _string;
```

```vb
Private _boolean As Boolean
```

```csharp
private bool _boolean;
```

```vb
Private _date As Date
```

```csharp
private DateTime _date;
```

```vb
Private _integer As Integer
```

```csharp
private int _integer;
```

```vb
Private _uinteger As UInteger
```

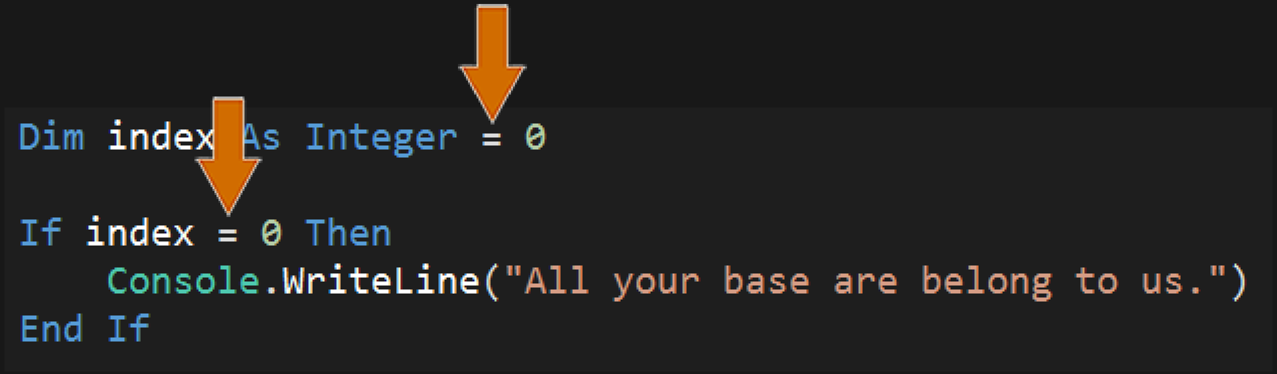
```csharp
private uint _uinteger;
```

```
te float _single;
```

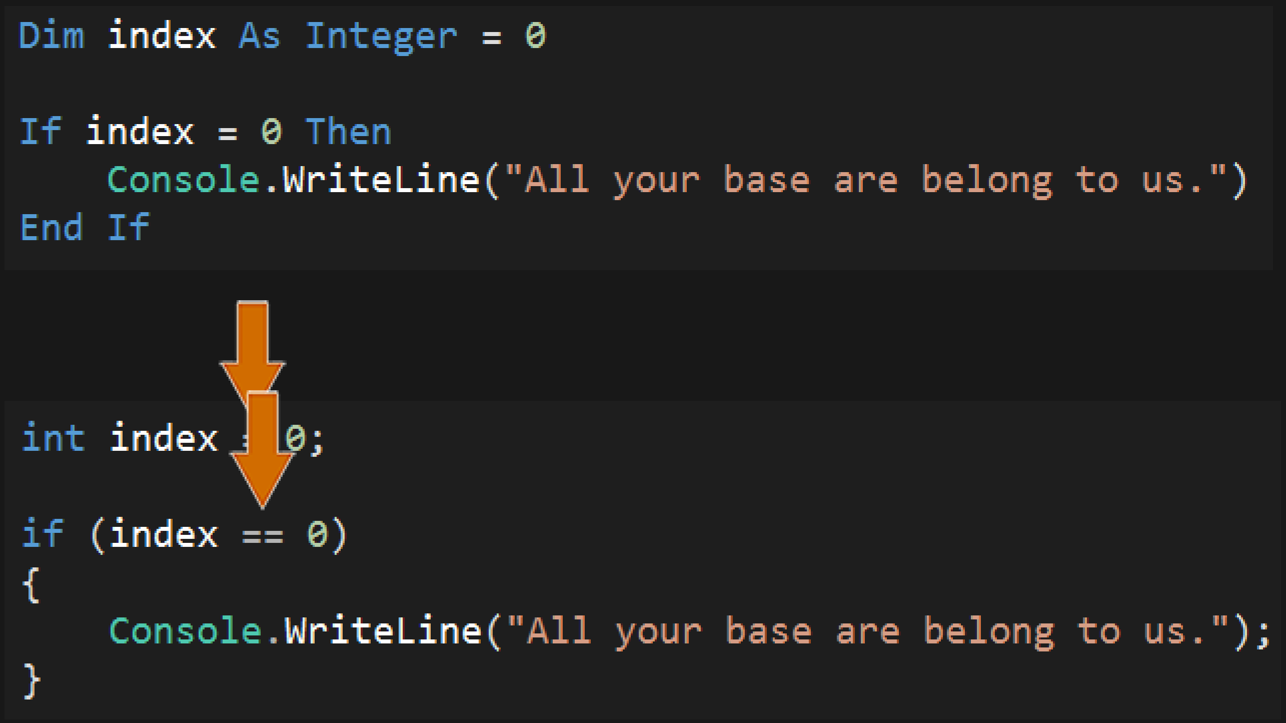
System.Single

Represents a single-precision floating-point number.

http://bit.ly/16LJDL3

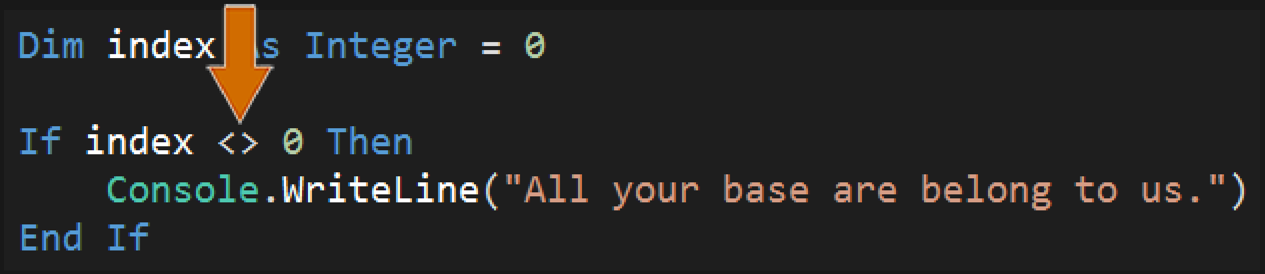


```
Dim index As Integer = 0

If index = 0 Then
    Console.WriteLine("All your base are belong to us.")
End If
```



```
Dim index As Integer = 0

If index = 0 Then
    Console.WriteLine("All your base are belong to us.")
End If
```

```
int index = 0;

if (index == 0)
{
    Console.WriteLine("All your base are belong to us.");
}
```



```
Dim index As Integer = 0

If index <> 0 Then
    Console.WriteLine("All your base are belong to us.")
End If
```

```vb
Dim index As Integer = 0

If index <> 0 Then
    Console.WriteLine("All your base are belong to us.")
End If
```

```csharp
int index = 0;

if (index != 0)
{
    Console.WriteLine("All your base are belong to us.");
```

The comparison operators:

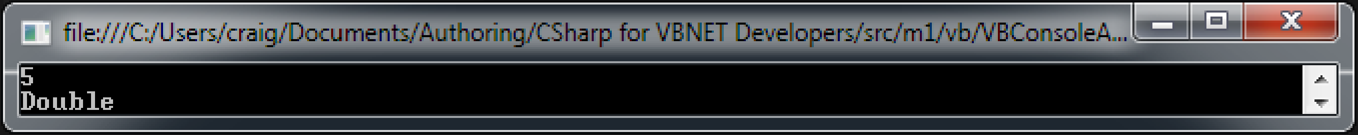
```
> >= < <=
```

and arithmetic operators:

```
+ - *
```

are the same among VB and C#.

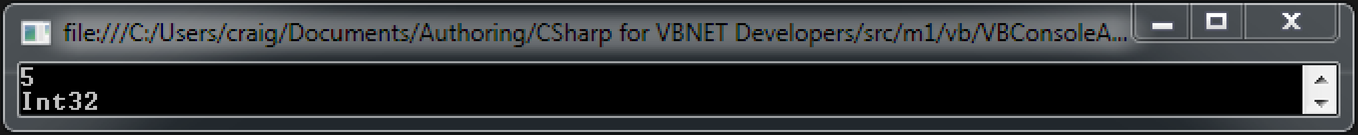
```
Dim result = (10 / 2)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Double
```

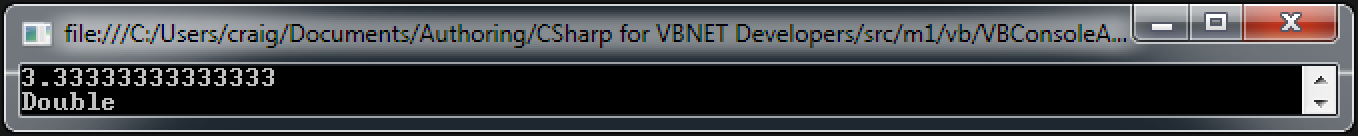
```
Dim result = (10 \ 2)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Int32
```

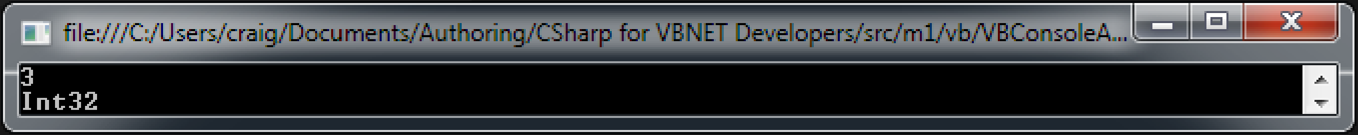
```
Dim result = (10 / 3)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

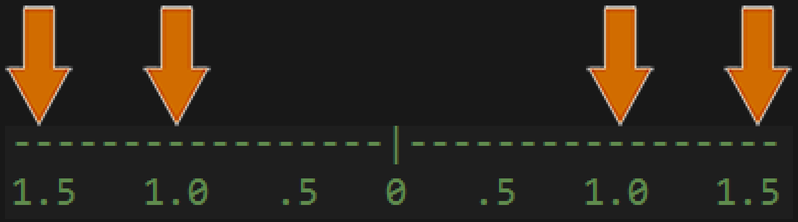
```
3.33333333333333
Double
```

```
Dim result = (10 \ 3)
```

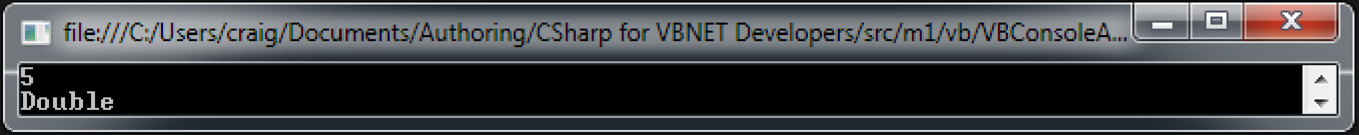


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
3
Int32
```


1.5
1.0
.5
0
.5
1.0
1.5

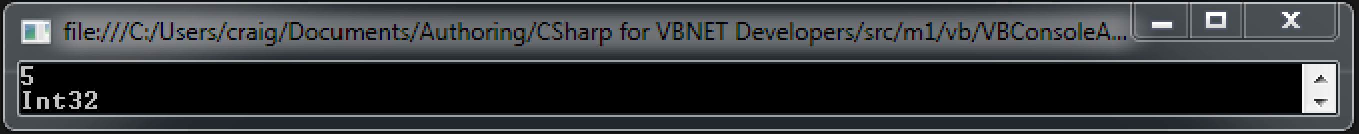
```
Dim result = (10 / 2)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Double
```

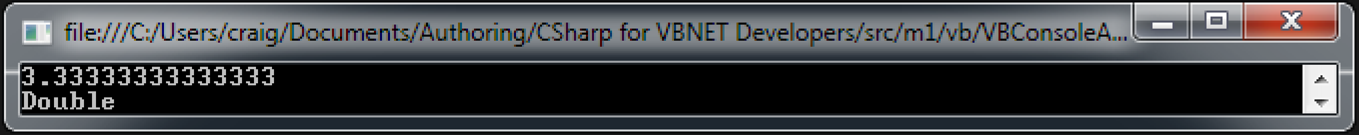
```
Dim result = (10 \ 2)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Int32
```

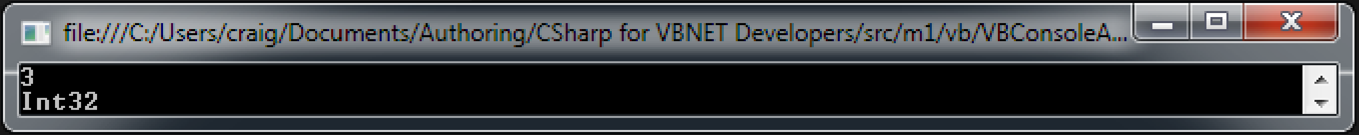
```
Dim result = (10 / 3)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
3.33333333333333
Double
```

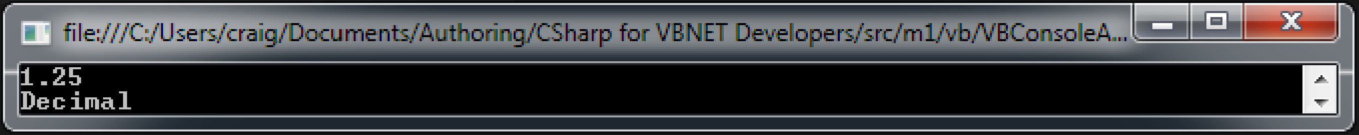
```
Dim result = (10 \ 3)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
3
Int32
```

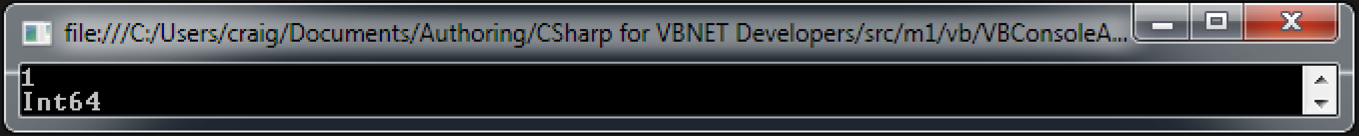
```
Dim result = (2.5D / 2)
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
1.25
Decimal
```

```
Dim result = (2.5D \ 2)
```

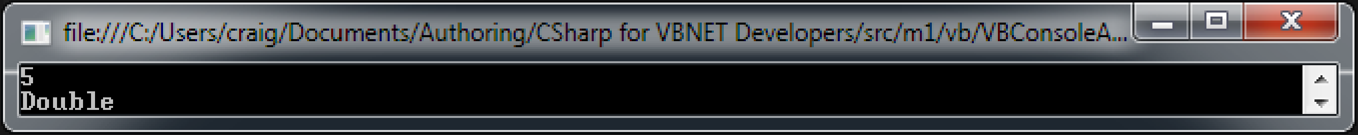


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
1
Int64
```

```
Dim result = (10 / 2)
```

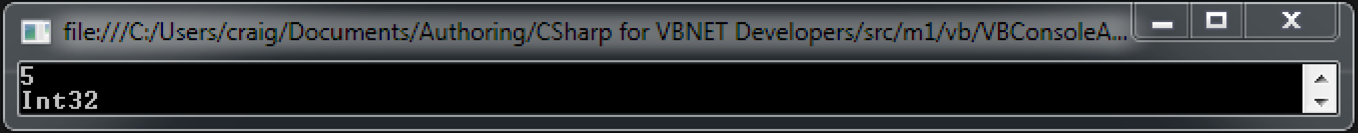
```
var result = (10d / 2d);
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Double
```

```
Dim result = (10 \ 2)
```

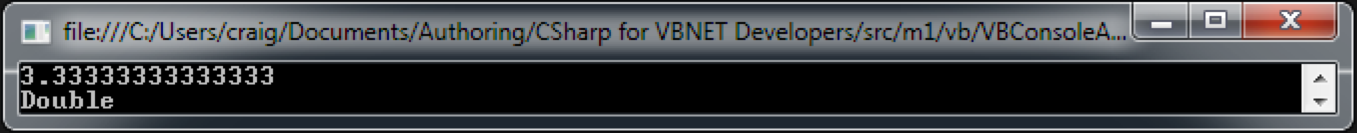
```
var result = (10 / 2);
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
5
Int32
```

```
Dim result = (10 / 3)
```

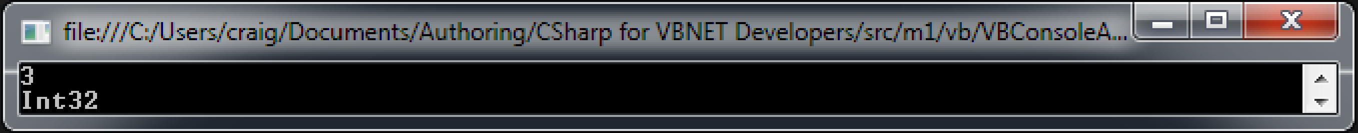
```
var result = (10d / 3d);
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
3.33333333333333
Double
```

```
Dim result = (10 \ 3)
```

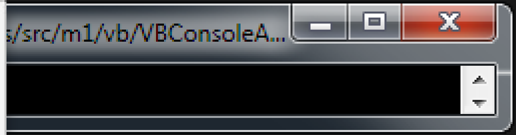
```
var result = (10 / 3);
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

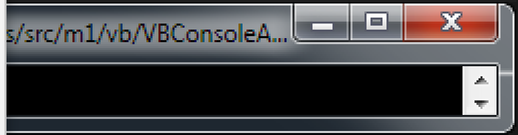
```
3
Int32
```

```
Dim result = (2.5D / 2)
```

```
var result = (2.5d / 2);
```

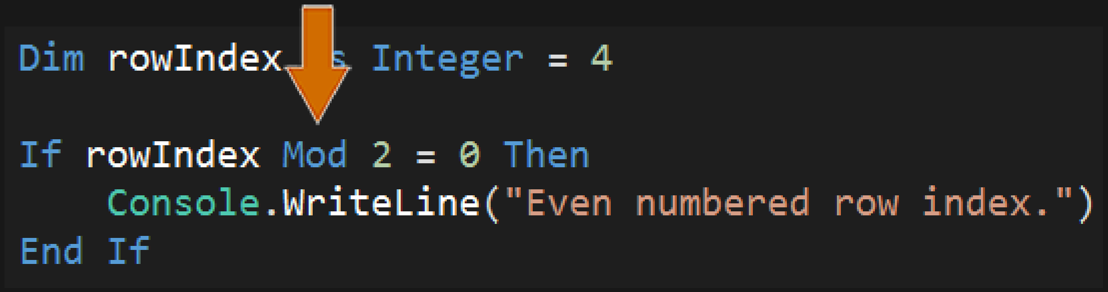


```
t = (((Int64)2.5d)/ 2);
```

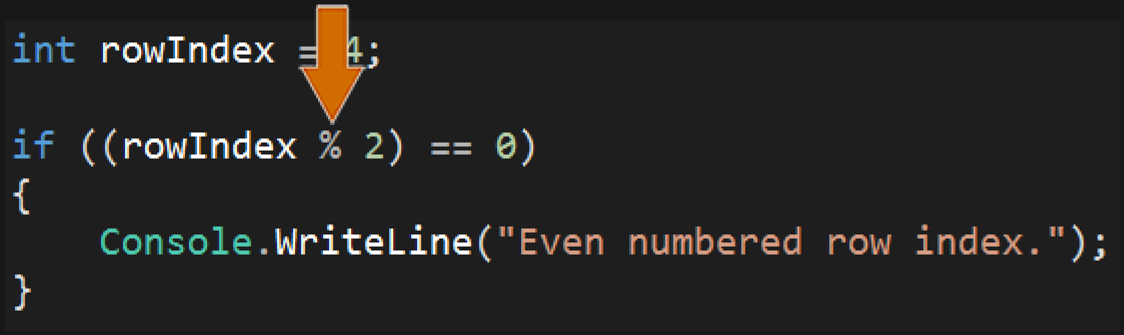


Tip

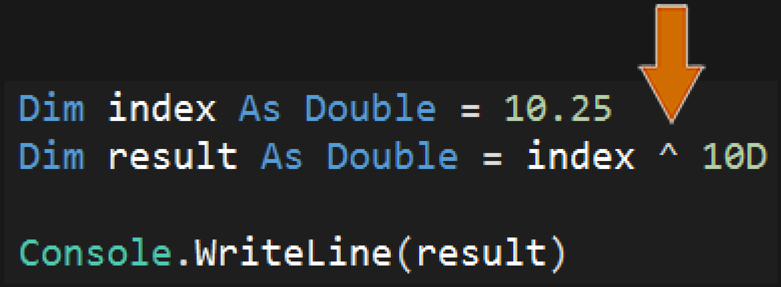
Try to use the `TryParse` method rather than a direct cast to avoid exceptions.



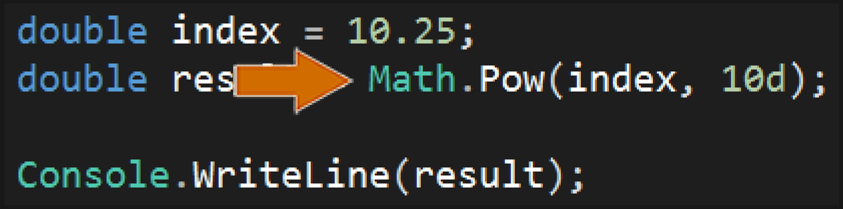
```vb
Dim rowIndex As Integer = 4

If rowIndex Mod 2 = 0 Then
    Console.WriteLine("Even numbered row index.")
End If
```



```csharp
int rowIndex = 4;

if ((rowIndex % 2) == 0)
{
    Console.WriteLine("Even numbered row index.");
}
```



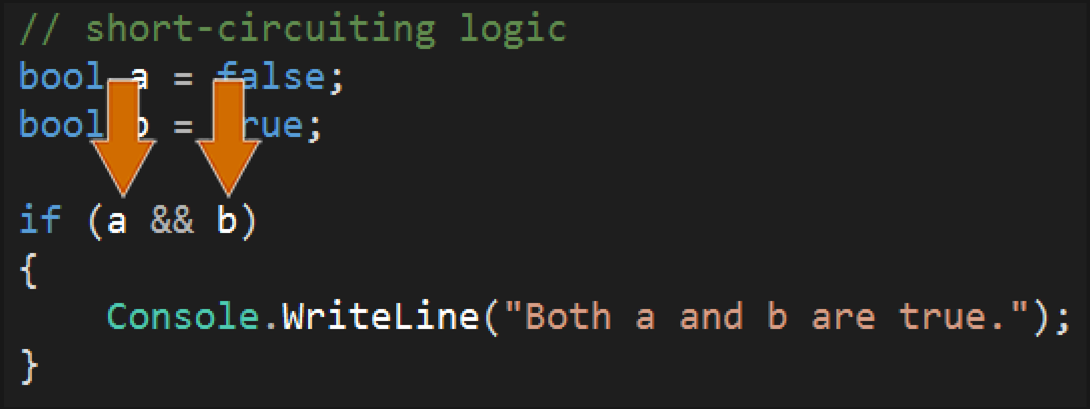
```
Dim index As Double = 10.25
Dim result As Double = index ^ 10D

Console.WriteLine(result)
```



```
double index = 10.25;
double res          Math.Pow(index, 10d);

Console.WriteLine(result);
```

```vb
If True AndAlso True Then
    Console.WriteLine("Both values are True.")
End If
```

```csharp
if (true && true)
{
    Console.WriteLine("Both values are true.");
}
```



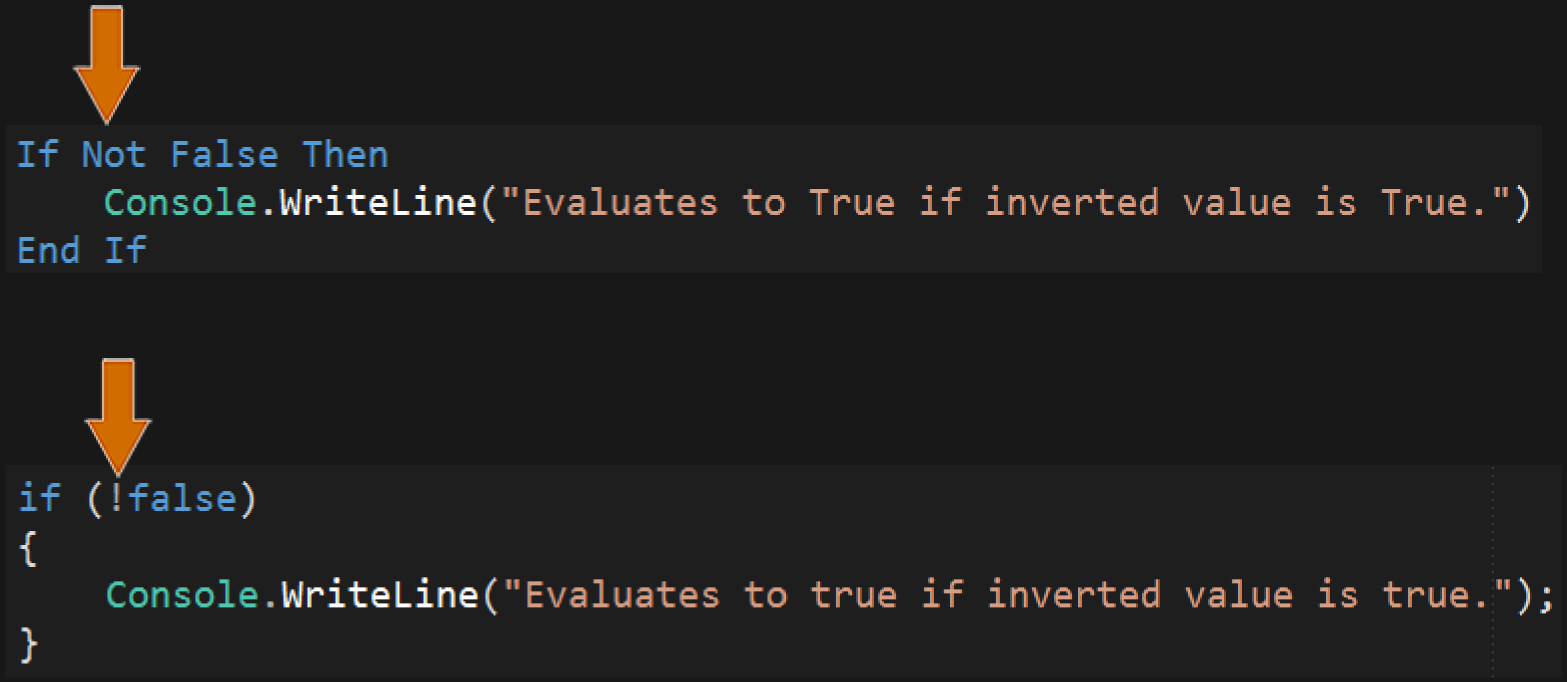
```
// short-circuiting logic
bool a = false;
bool b = true;

if (a && b)
{
    Console.WriteLine("Both a and b are true.");
}
```

```
If False OrElse True Then
    Console.WriteLine("At least one value is True.")
End If
```

```
e value is true.");
```

Other Operator Equivalents:

VB.NET	C#
And	&
Or	|
Xor	^
Not	~
<<	<<
>>	>>



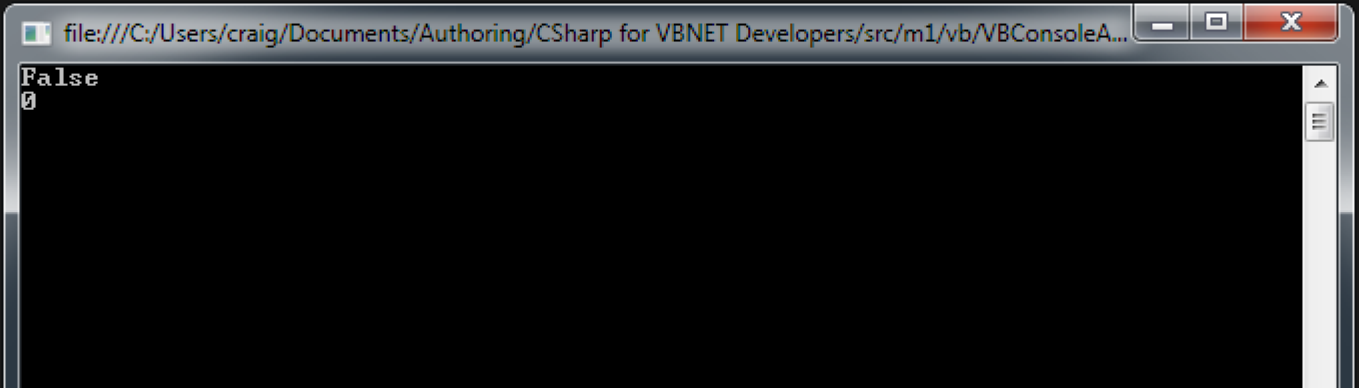
```
If Not False Then
    Console.WriteLine("Evaluates to True if inverted value is True.")
End If
```

```
if (!false)
{
    Console.WriteLine("Evaluates to true if inverted value is true.");
}
```

```vb
Dim rootPath As String

If ConfigurationManager.AppSettings("rootPath") IsNot Nothing Then
    rootPath = ConfigurationManager.AppSettings("appSettings").ToString()
End If
```

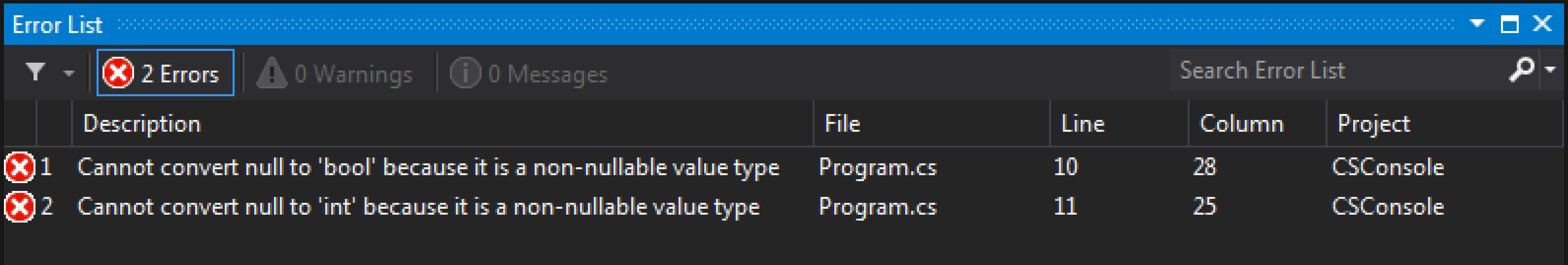
```csharp
string rootPath;

if (ConfigurationManager.AppSettings["rootPath"] != null)
{
    rootPath = ConfigurationManager.AppSettings["rootPath"].ToString();
}
```

```vb
Dim isReady As Boolean = Nothing
Dim index As Integer = Nothing

Console.WriteLine(isReady)
Console.WriteLine(index)
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

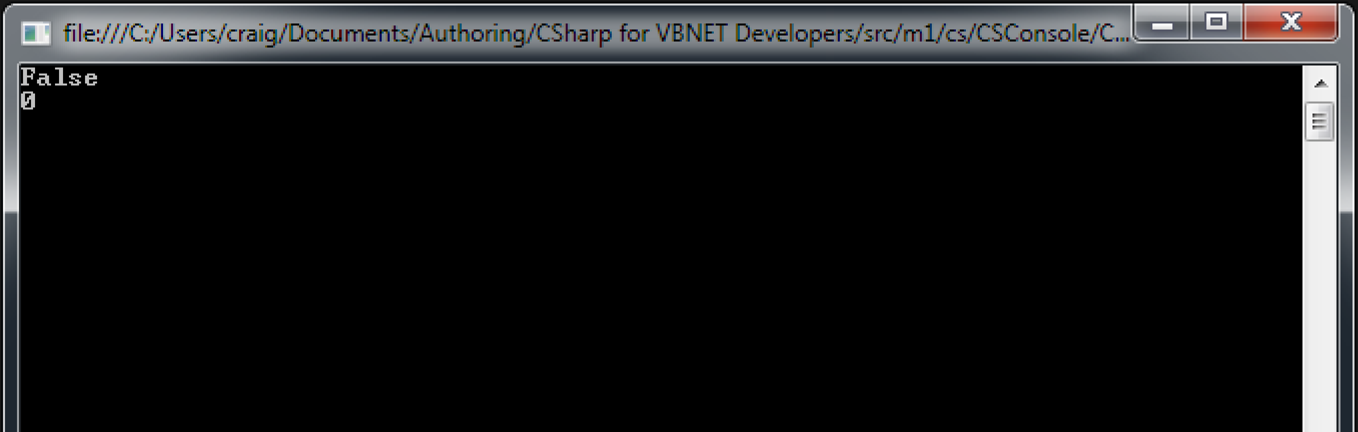
```
False
0
```

```
bool isReady = null;
int index = null;

Console.WriteLine(isReady);
Console.WriteLine(index);
```



	Description	File	Line	Column	Project
1	Cannot convert null to 'bool' because it is a non-nullable value type	Program.cs	10	28	CSConsole
2	Cannot convert null to 'int' because it is a non-nullable value type	Program.cs	11	25	CSConsole

```
bool isReady = default(bool);
int index = default(int);

Console.WriteLine(isReady);
Console.WriteLine(index);
```



file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/cs/CSConsole/C...

```
False
0
```

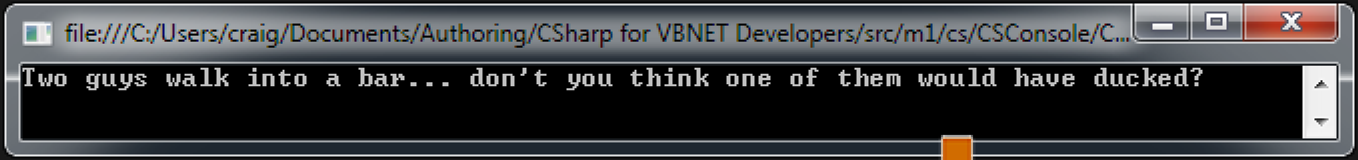
```vb
Enum StatusTypes
    Unknown = -1
    Started = 1
    InProcess
    Complete
    Ended = Complete
    Rejected
End Enum
```

```csharp
enum StatusTypes
{
    Unknown = -1,
    Started = 1,
    InProcess,
    Complete,
    Ended = Complete,
    Rejected
}
```

abstract	event	new	struct
as	explicit	null	switch
base	extern	object	this
bool	false	operator	throw
break	finally	out	true
byte	fixed	override	try
case	float	params	typeof
catch	for	private	uint
char	foreach	protected	ulong
checked	goto	public	unchecked
class	if	readonly	unsafe
const	implicit	ref	ushort
continue	in	return	using
decimal	int	sbyte	virtual
default	interface	sealed	void
delegate	internal	short	volatile
do	is	sizeof	while
double	lock	stackalloc	
else	long	static	
enum	namespace	string	

abstract	float	override	this
base	foreach	params	uint
bool	implicit	ref	unchecked
break	int	sealed	unsafe
checked	internal	sizeof	virtual
explicit	lock	stackalloc	void
extern	null	struct	volatile
fixed	out	switch	

```
Console.WriteLine("Two guys walk into a bar..." &
                  " don't you think one of them would have ducked?")
```

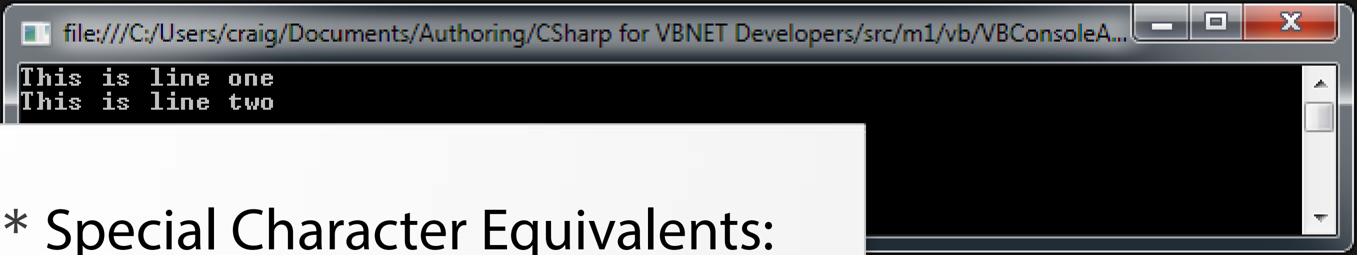


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/cs/CSConsole/C...

```
Two guys walk into a bar... don't you think one of them would have ducked?
```

```
Console.WriteLine("Two guys walk into a bar..." +
                  " don't you think one of them would have ducked?");
```

```
Console.WriteLine("This is line one " & vbCrLf & "This is line two")
Console.WriteLine()
Console.WriteLine("This is line one " & ControlChars.CrLf & "This is line two")
```



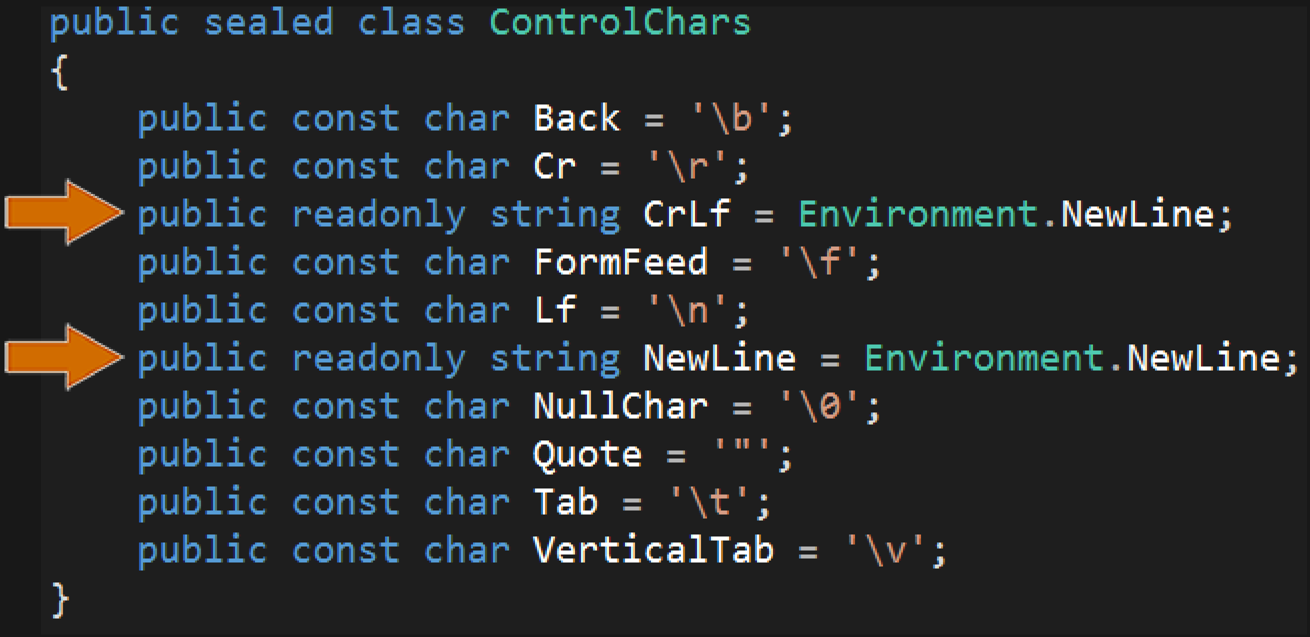
file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/vb/VBConsoleA...

```
This is line one
This is line two
```

```
(
ine two");
```

vb* Special Character Equivalents:

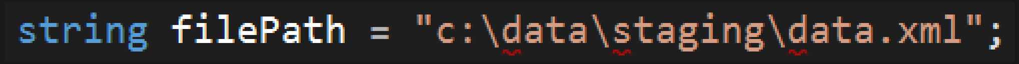
VB.NET	C#
vbCrLf	\r\n
vbNewLine	\r\b
vbCr	\r
vbLf	\n
vbBack	\b
vbFormFeed	\f
vbVerticalTab	\v



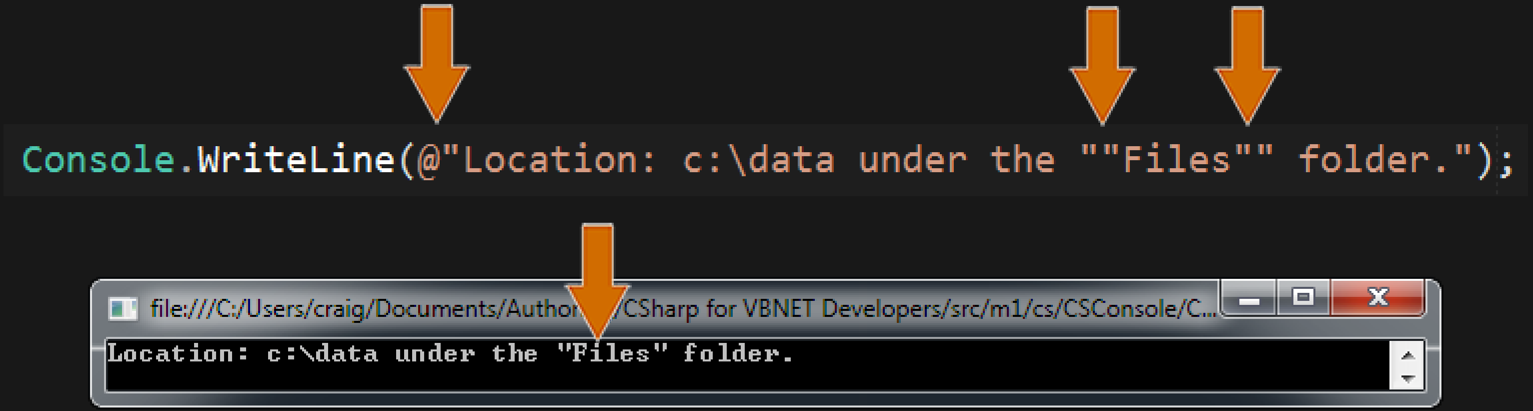
```
public sealed class ControlChars
{
    public const char Back = '\b';
    public const char Cr = '\r';
    public readonly string CrLf = Environment.NewLine;
    public const char FormFeed = '\f';
    public const char Lf = '\n';
    public readonly string NewLine = Environment.NewLine;
    public const char NullChar = '\0';
    public const char Quote = '"';
    public const char Tab = '\t';
    public const char VerticalTab = '\v';
}
```

```
If "info@pluralsight.com" Like "*@pluralsight.com" Then
    Console.WriteLine("Pluralsight email address.")
End If
```

```
if ("info@pluralsight.com".Contains("@pluralsight.com"))
{
    Console.WriteLine("Pluralsight email address.");
}
```



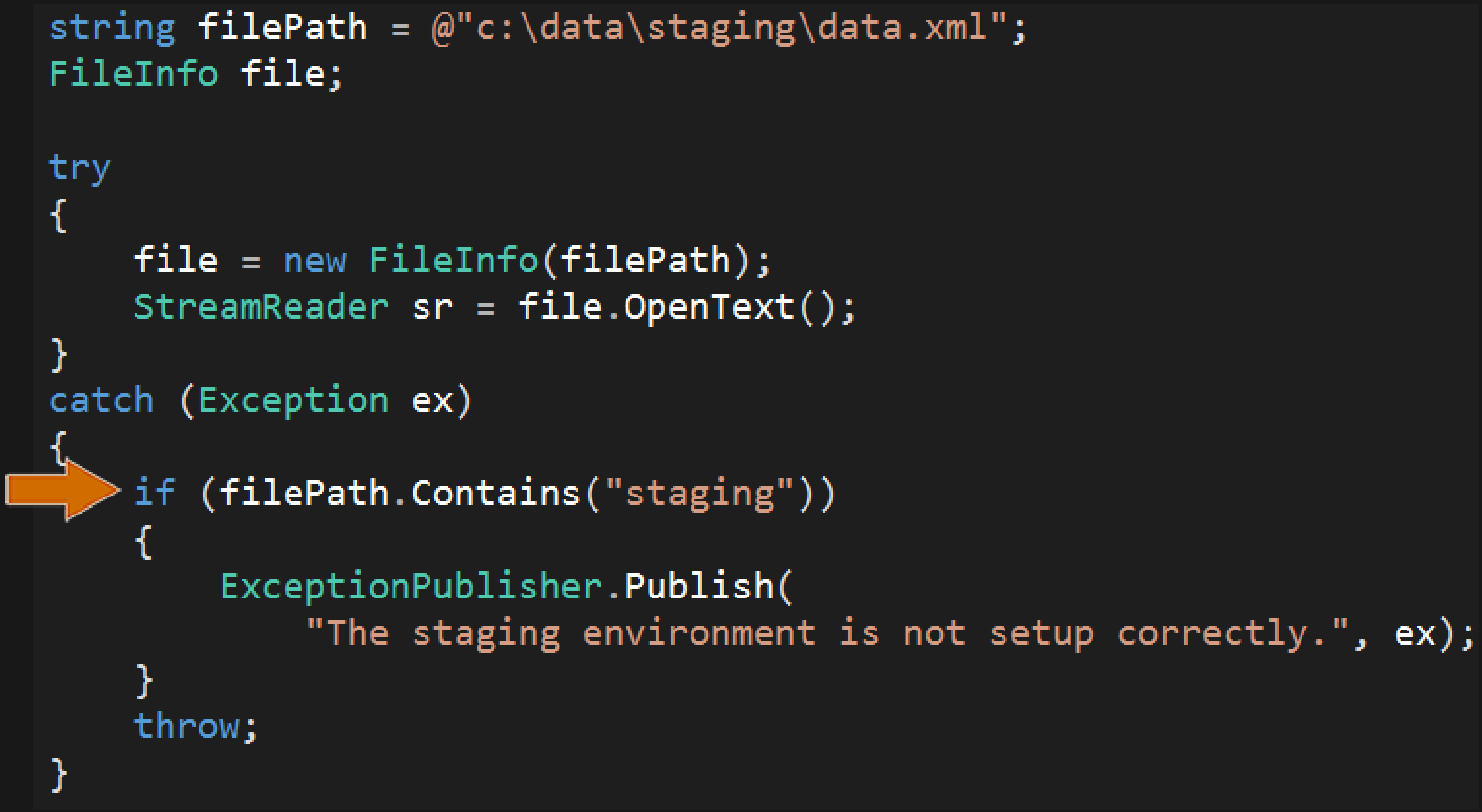
```
string filePath = "c:\data\staging\data.xml";
string filePath = "c:\\data\\staging\\data.xml";
string filePath = @"c:\data\staging\data.xml";
```


```csharp
Console.WriteLine(@"Location: c:\data under the ""Files"" folder.");
```

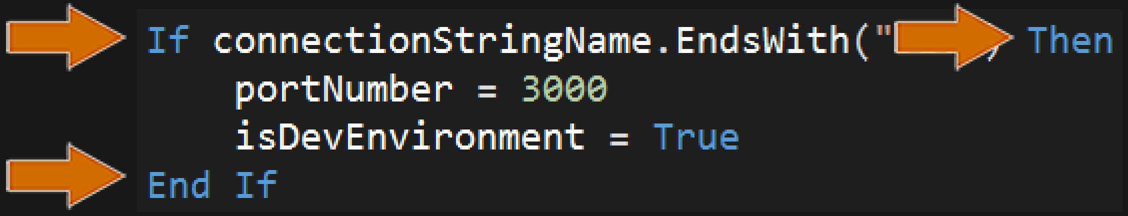
file:///C:/Users/craig/Documents/Author.../CSharp for VBNET Developers/src/m1/cs/CSConsole/C...

```
Location: c:\data under the "Files" folder.
```

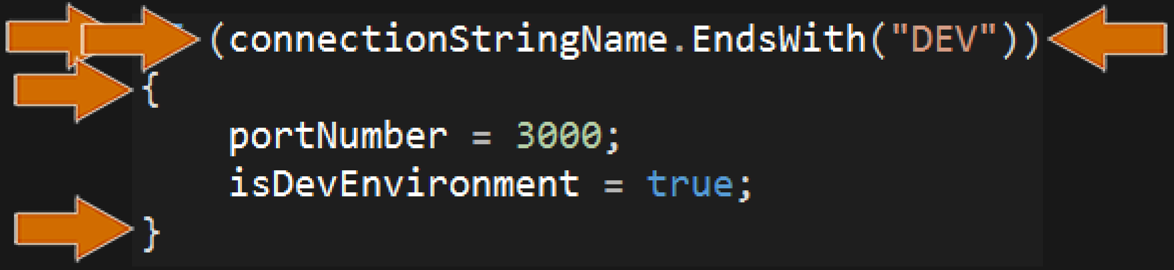
```vb
Dim filePath As String = "c:\data\staging\data.xml"
Dim file As FileInfo

Try
    file = New FileInfo(filePath)
    Dim sr As StreamReader = file.OpenText()
Catch ex As Exception When filePath.Contains("staging")
    ExceptionPublisher.Publish(
        "The staging environment is not setup correctly.", ex)
    Throw
End Try
```



```csharp
string filePath = @"c:\data\staging\data.xml";
FileInfo file;

try
{
    file = new FileInfo(filePath);
    StreamReader sr = file.OpenText();
}
catch (Exception ex)
{
    if (filePath.Contains("staging"))
    {
        ExceptionPublisher.Publish(
            "The staging environment is not setup correctly.", ex);
    }
    throw;
}
```



```
If connectionStringName.EndsWith("       Then
    portNumber = 3000
    isDevEnvironment = True
End If
```



```
   (connectionStringName.EndsWith("DEV"))
{
    portNumber = 3000;
    isDevEnvironment = true;
}
```

Tip

Always use curly braces following `if` statements.

```vb
If connectionStringName Is Nothing Then
    connectionStringName = "DefaultConnection"
    isDevEnvironment = True
    portNumber = 3000
ElseIf connectionStringName.EndsWith("DEV") Then
    portNumber = 3000
    isDevEnvironment = True
End If
```

```csharp
if (connectionStringName == null)
{
    connectionStringName = "DefaultConnection";
    isDevEnvironment = true;
    portNumber = 3000;
}
else if (connectionStringName.EndsWith("DEV"))
{
    portNumber = 3000;
    isDevEnvironment = true;
}
```

```vb
If connectionStringName IsNot Nothing AndAlso connectionStringName.EndsWith("DEV") Then
    portNumber = 3000
End If
```

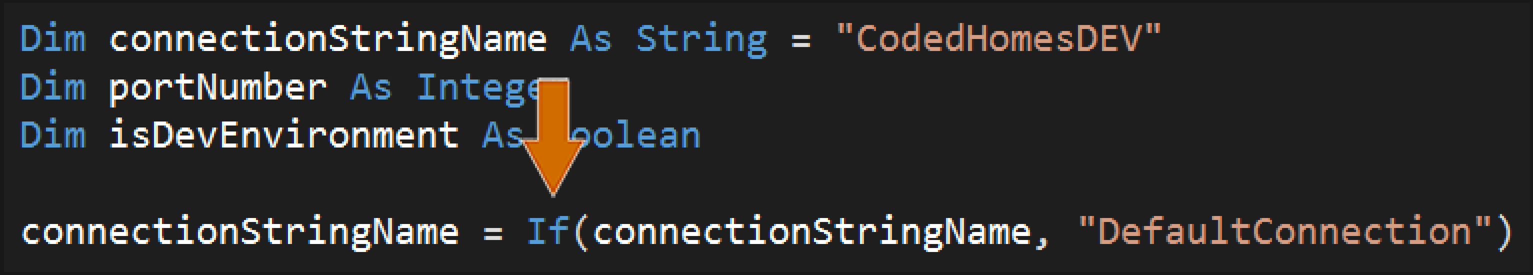
```csharp
if (connectionStringName != null && connectionStringName.EndsWith("DEV"))
{
    portNumber = 3000;
}
```

Tip

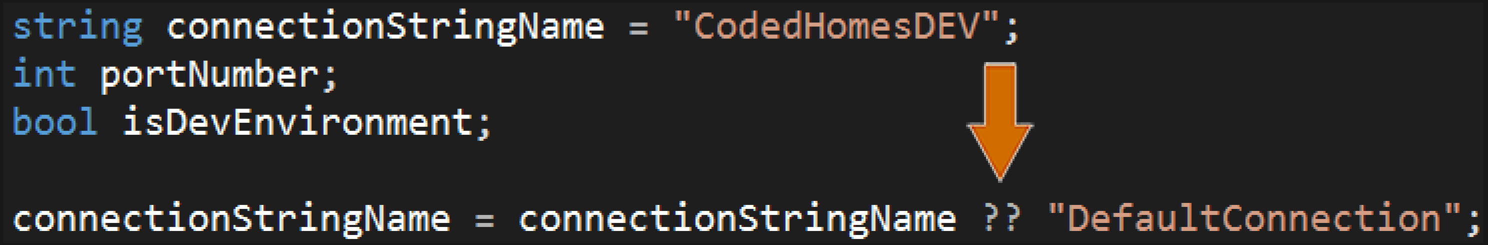
Add parenthesis to your expression
to enforce order of operations.

```vb
If connectionStringName IsNot Nothing _
    AndAlso connectionStringName.EndsWith("DEV") Then
    portNumber = 3000
End If
```

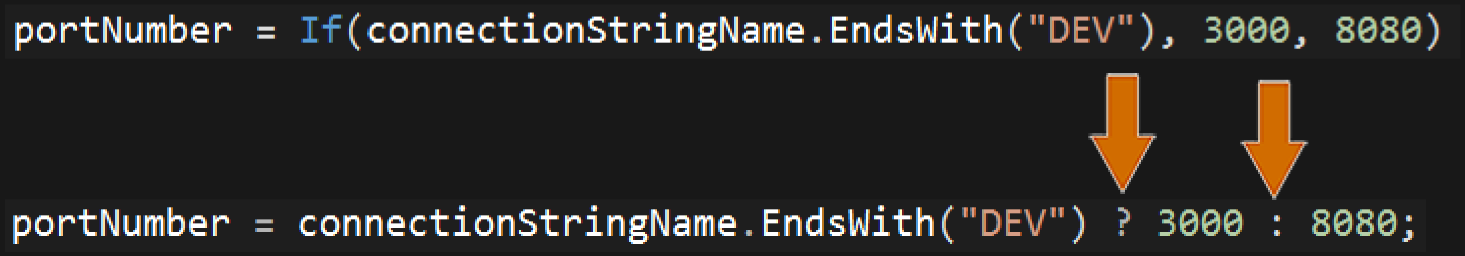
```csharp
if (connectionStringName != null
    && connectionStringName.EndsWith("DEV"))
{
    portNumber = 3000;
}
```



```vb
Dim connectionStringName As String = "CodedHomesDEV"
Dim portNumber As Integer
Dim isDevEnvironment As Boolean

connectionStringName = If(connectionStringName, "DefaultConnection")
```



```csharp
string connectionStringName = "CodedHomesDEV";
int portNumber;
bool isDevEnvironment;

connectionStringName = connectionStringName ?? "DefaultConnection";
```



```
portNumber = If(connectionStringName.EndsWith("DEV"), 3000, 8080)
```

```
portNumber = connectionStringName.EndsWith("DEV") ? 3000 : 8080;
```

```
If connectionStringName.EndsWith("DEV") Then portNumber = 3000
```

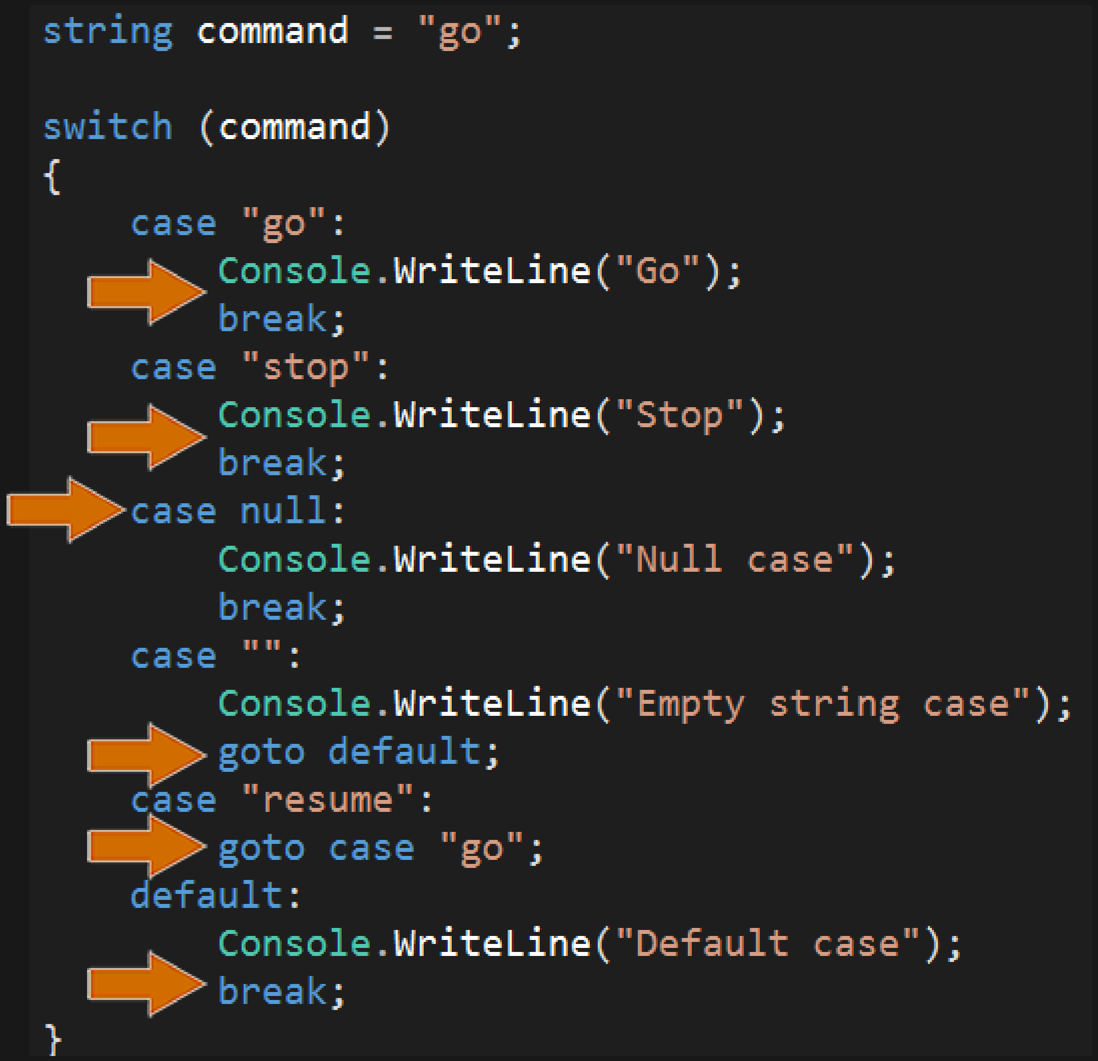
```
if (connectionStringName.EndsWith("DEV")) portNumber = 3000;
```

```vb
If connectionStringName.EndsWith("DEV") Then portNumber = 3000 : isDevEnvironment = True
```

```csharp
if (connectionStringName.EndsWith("DEV"))
{
    portNumber = 3000;
    isDevEnvironment = true;
}
```

```vb
Select Case lowInventoryThreshold
    Case 100
        InventoryUnit.Order(200)
    Case 250
        InventoryUnit.Order(250)
    Case 500
        InventoryUnit.Order(500)
    Case Else
        InventoryUnit.Order(10)
End Select
```

```csharp
switch (lowInventoryThreshold)
{
    case 100:
        InventoryUnit.Order(200);
        break;
    case 250:
        InventoryUnit.Order(250);
        break;
    case 500:
        InventoryUnit.Order(500);
        break;
    default:
        InventoryUnit.Order(10);
        break;
}
```



```
string command = "go";

switch (command)
{
    case "go":
        Console.WriteLine("Go");
        break;
    case "stop":
        Console.WriteLine("Stop");
        break;
    case null:
        Console.WriteLine("Null case");
        break;
    case "":
        Console.WriteLine("Empty string case");
        goto default;
    case "resume":
        goto case "go";
    default:
        Console.WriteLine("Default case");
        break;
}
```

```
While count > 0
    ' do something interesting
    count -= 1
End While
```

```
Do While count > 0
    ' do something interesting
    count -= 1
Loop
```

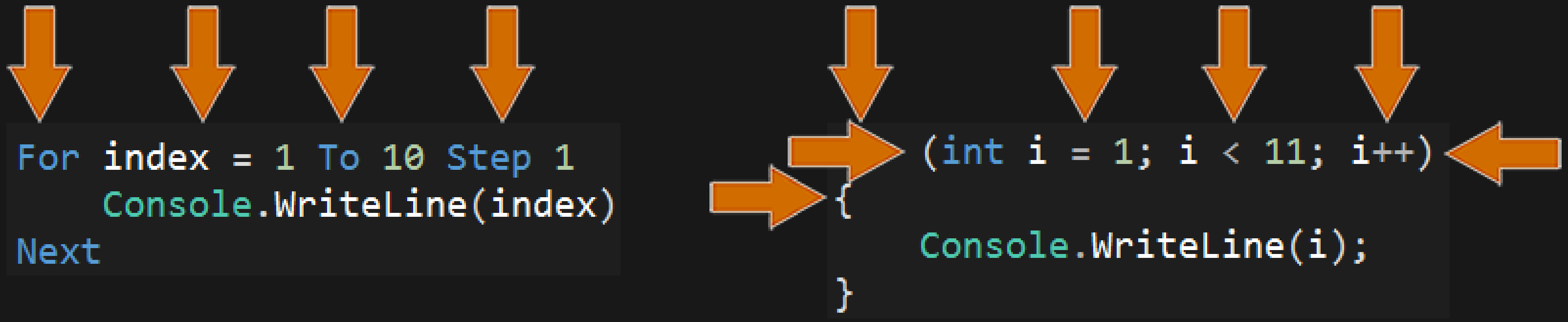
```
Do Until count = 1
    ' do something interesting
    count -= 1
Loop
```

```
while (count > 0)
{
    // do something interesting
    count -= 1;
}
```

```
do
{
    // do something interesting
    count -= 1;
} while (count > 0);
```

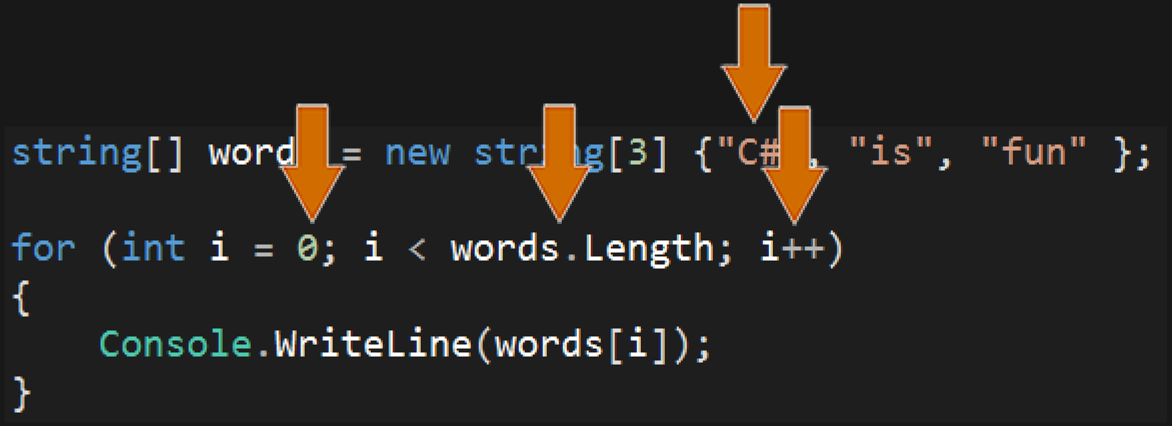
```
Do Until count = 1
    ' do something interesting
    count -= 1
Loop
```

```
do
{
    // do something interesting
    count -= 1;
} while (count != 1);
```



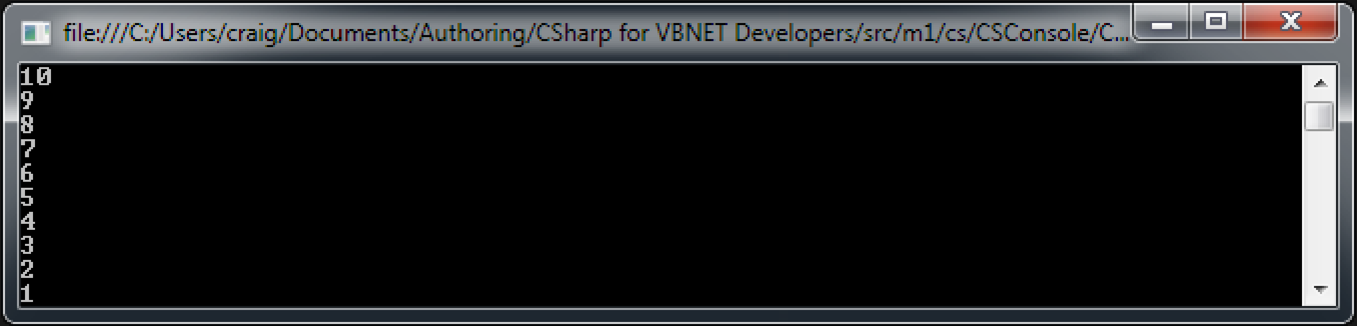
```vb
For index = 1 To 10 Step 1
    Console.WriteLine(index)
Next
```

```csharp
 (int i = 1; i < 11; i++)
{
    Console.WriteLine(i);
}
```

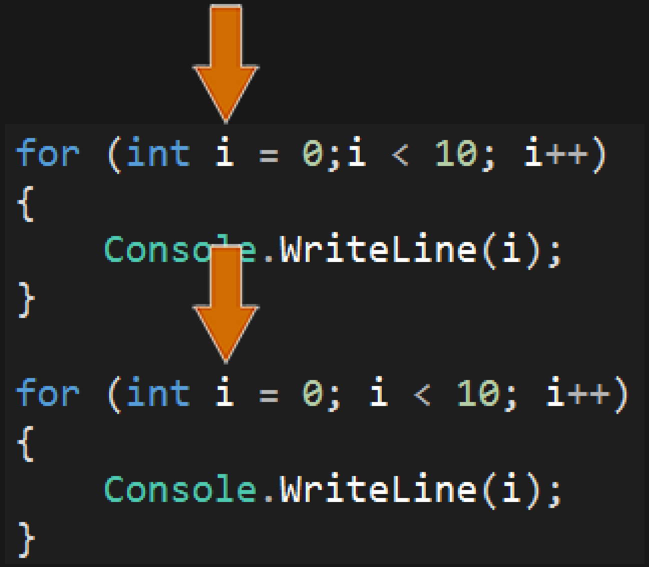


```
string[] words = new string[3] {"C#", "is", "fun" };

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

```
for (int i = 10; i > 0; i--)
{
    Console.WriteLine(i);
}
```


file:///C:/Users/craig/Documents/Authoring/CSharp for VBNET Developers/src/m1/cs/CSConsole/C...

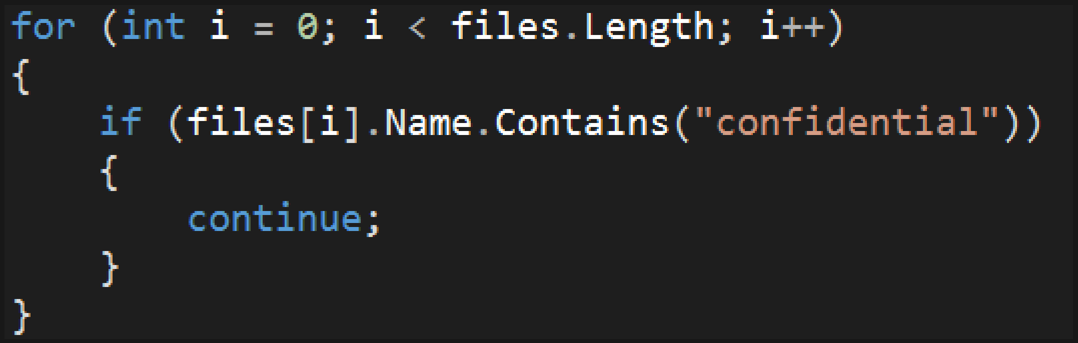
```
10
9
8
7
6
5
4
3
2
1
```



```
for (int i = 0;i < 10; i++)
{
    Console.WriteLine(i);
}

for (int i = 0; i < 10; i++)
{
    Console.WriteLine(i);
}
```

```
For index = 0 To files.Length
    If files(index).Name.Contains("confidential") Then
        Exit For
    End If
Next
```

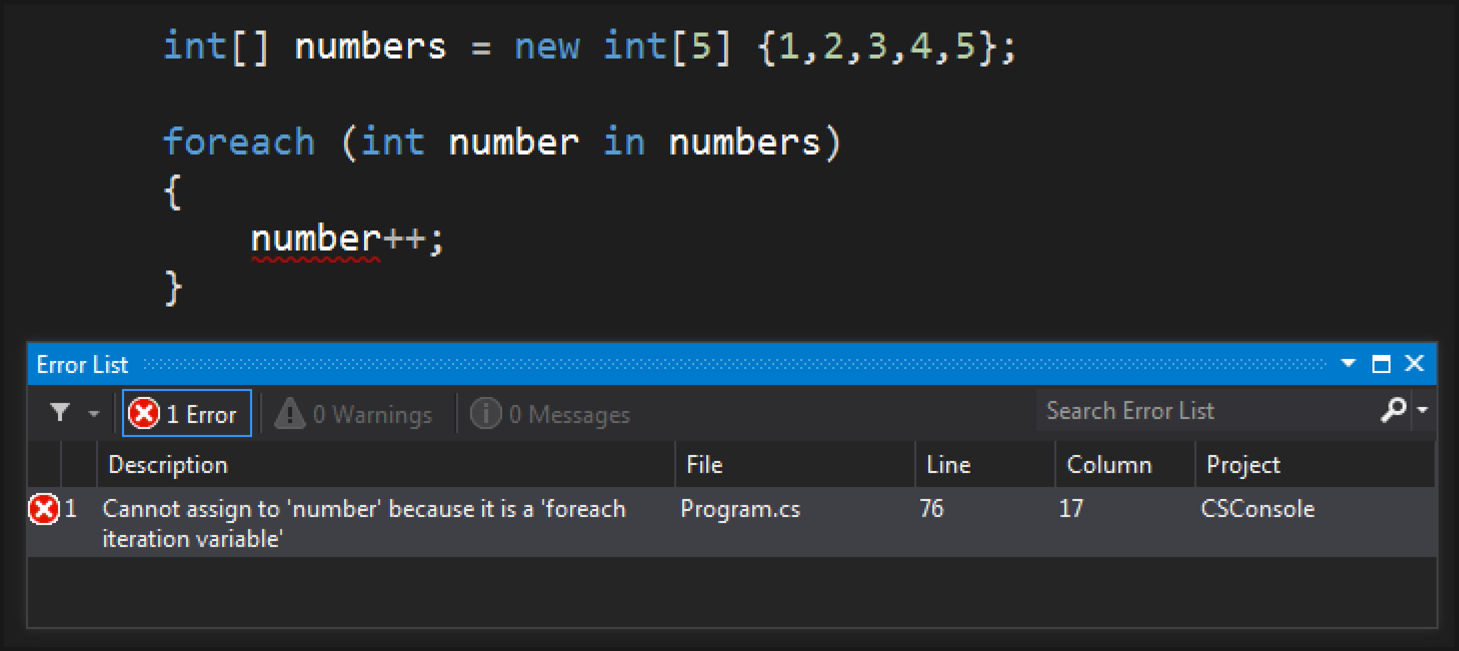
```
for (int i = 0; i < files.Length; i++)
{
    if (files[i].Name.Contains("confidential"))
    {
        break;
    }
}
```



```csharp
for (int i = 0; i < files.Length; i++)
{
    if (files[i].Name.Contains("confidential"))
    {
        continue;
    }
}
```

```
Dim directory As New DirectoryInfo("c:\data")

For Each file As FileInfo In directory.GetFiles()
    Console.WriteLine(file.Name)
Next
```

```
DirectoryInfo directory = new DirectoryInfo(@"c:\data");

foreach (FileInfo file in directory.GetFiles())
{
    Console.WriteLine(file.Name);
}
```

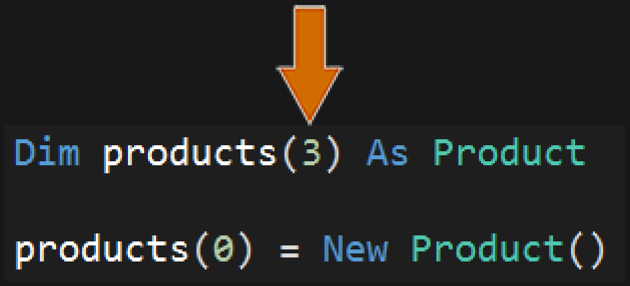


```
int[] numbers = new int[5] {1,2,3,4,5};

foreach (int number in numbers)
{
    number++;
}
```

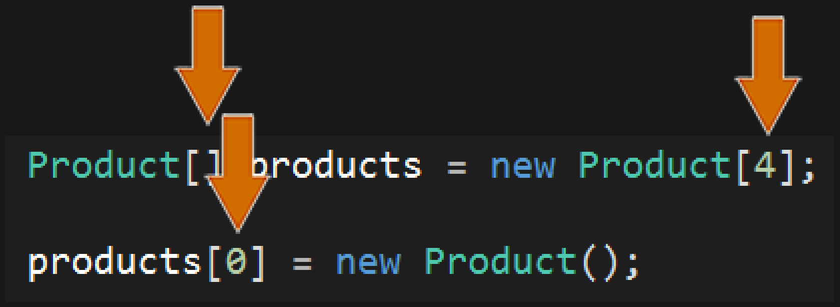
```
   Description                                              File         Line   Column   Project
1  Cannot assign to 'number' because it is a 'foreach      Program.cs   76     17       CSConsole
   iteration variable'
```

```vb
Dim product As New Product()

With product
    .Id = 1
    .Name = "Galactic Bounce Balls"
    .Description = "Super awesome bouncy balls."
    .QuantityOnHand = 10
    .QuantityOnOrder = 10
End With
```

```csharp
Product product = new Product();

product.Id = 1;
product.Name = "Galactic Bounce Balls";
product.Description = "Super awesome bouncy balls.";
product.QuantityOnHand = 10;
product.QuantityOnOrder = 10;
```



```vb
Dim products(3) As Product

products(0) = New Product()
```

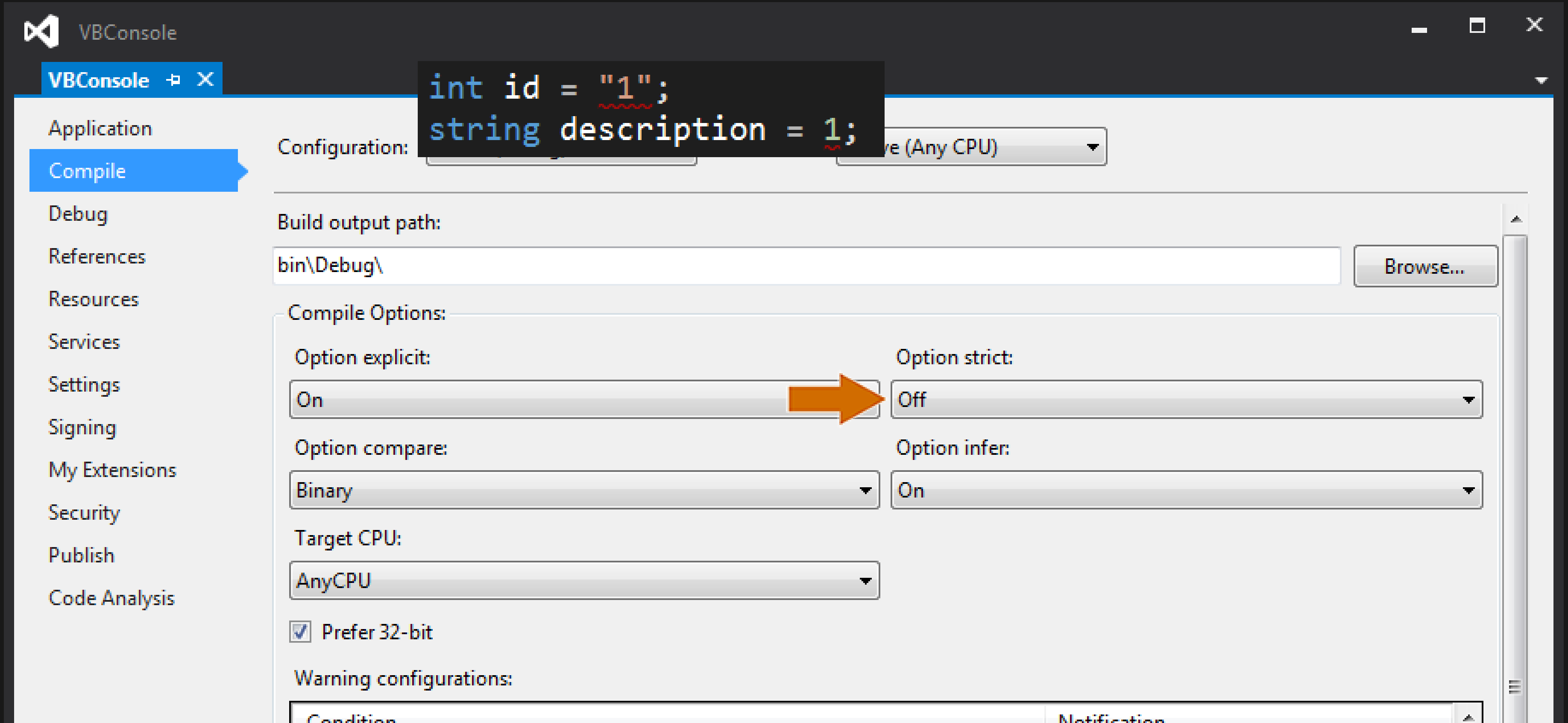


```csharp
Product[] products = new Product[4];

products[0] = new Product();
```

```vb
Dim products(3) As Product

products(0) = New Product()
products(1) = New Product()
products(2) = New Product()
products(3) = New Product()

ReDim Preserve products(5)

products(4) = New Product()
products(5) = New Product()
```

```csharp
Product[] products = new Product[4];

products[0] = new Product();
products[1] = new Product();
products[2] = new Product();
products[3] = new Product();

Array.Resize<Product>(ref products, 6);

products[4] = new Product();
products[5] = new Product();
```

```vb
Dim id As Integer = "1"
Dim description As String = 1
```



```csharp
int id = "1";
string description = 1;
```

VBConsole

Application
Compile
Debug
References
Resources
Services
Settings
Signing
My Extensions
Security
Publish
Code Analysis

Configuration: ve (Any CPU)

Build output path: bin\Debug\

Compile Options:

Option explicit: On

Option strict: Off

Option compare: Binary

Option infer: On

Target CPU: AnyCPU

[✓] Prefer 32-bit

Warning configurations:

Condition	Notification

```
Dim id As Integer = "1"
Dim description As String = 1
```

```
int id = 1;
string description = "1";
```

Summary

{ ; [

NEXT

Classes, Interfaces & Inheritance