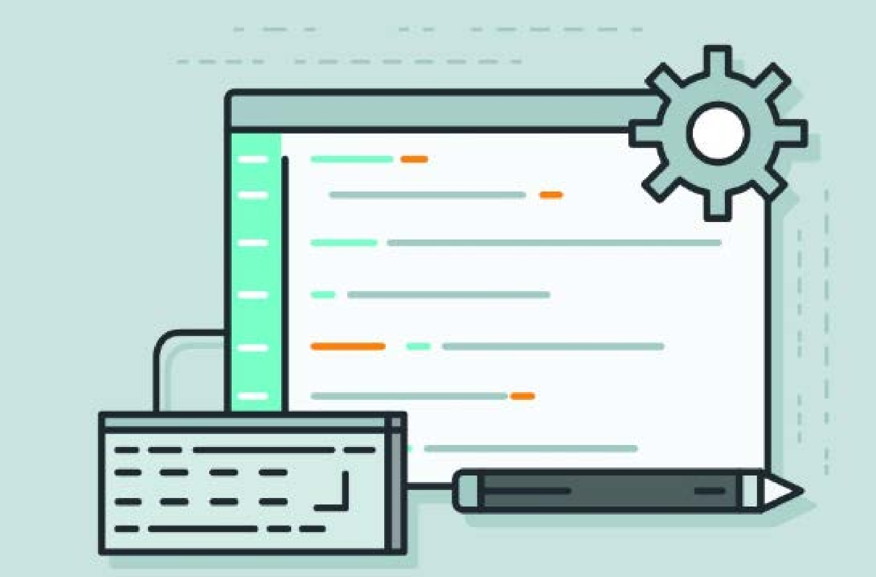
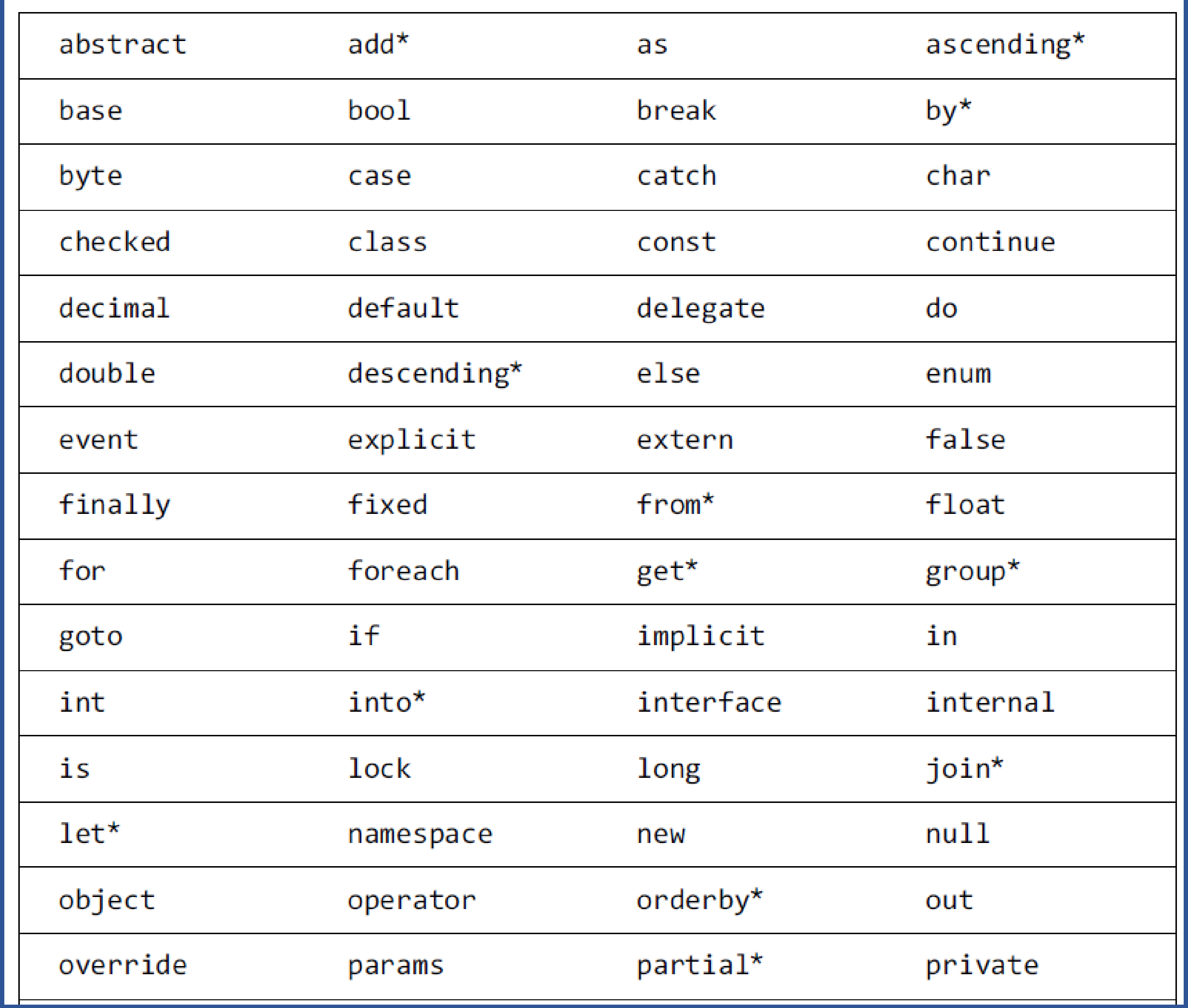
Data Type

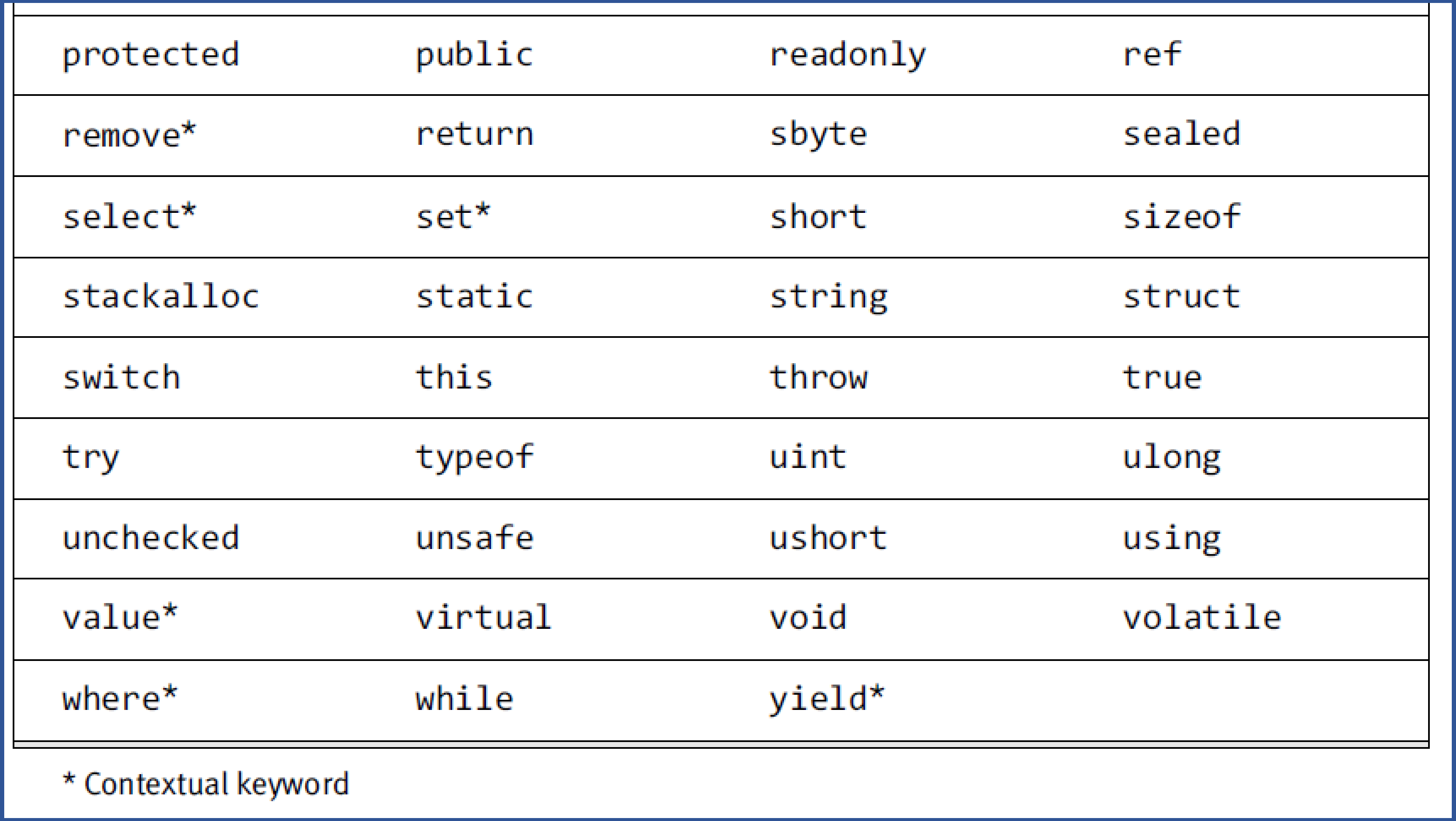


C# Keywords



abstract	add*	as	ascending*
base	bool	break	by*
byte	case	catch	char
checked	class	const	continue
decimal	default	delegate	do
double	descending*	else	enum
event	explicit	extern	false
finally	fixed	from*	float
for	foreach	get*	group*
goto	if	implicit	in
int	into*	interface	internal
is	lock	long	join*
let*	namespace	new	null
object	operator	orderby*	out
override	params	partial*	private

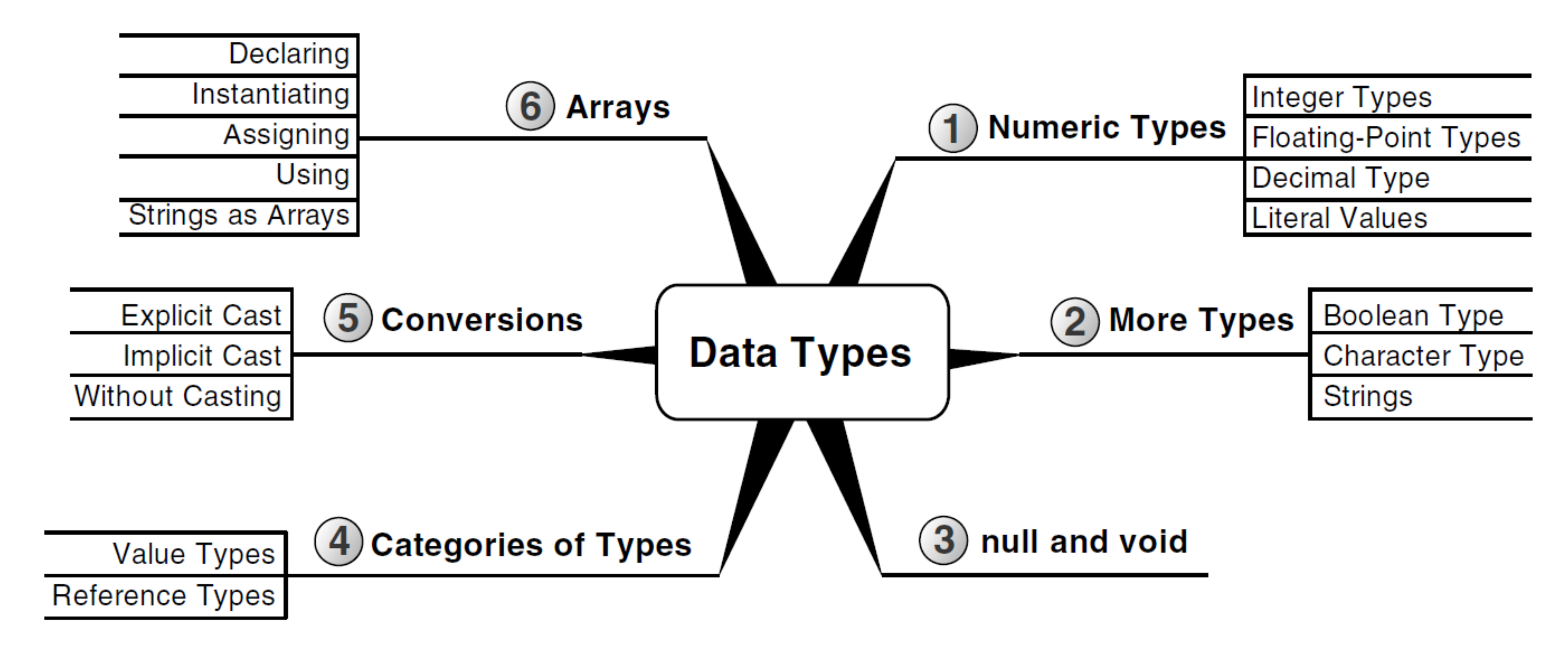
C# Keywords (cont')

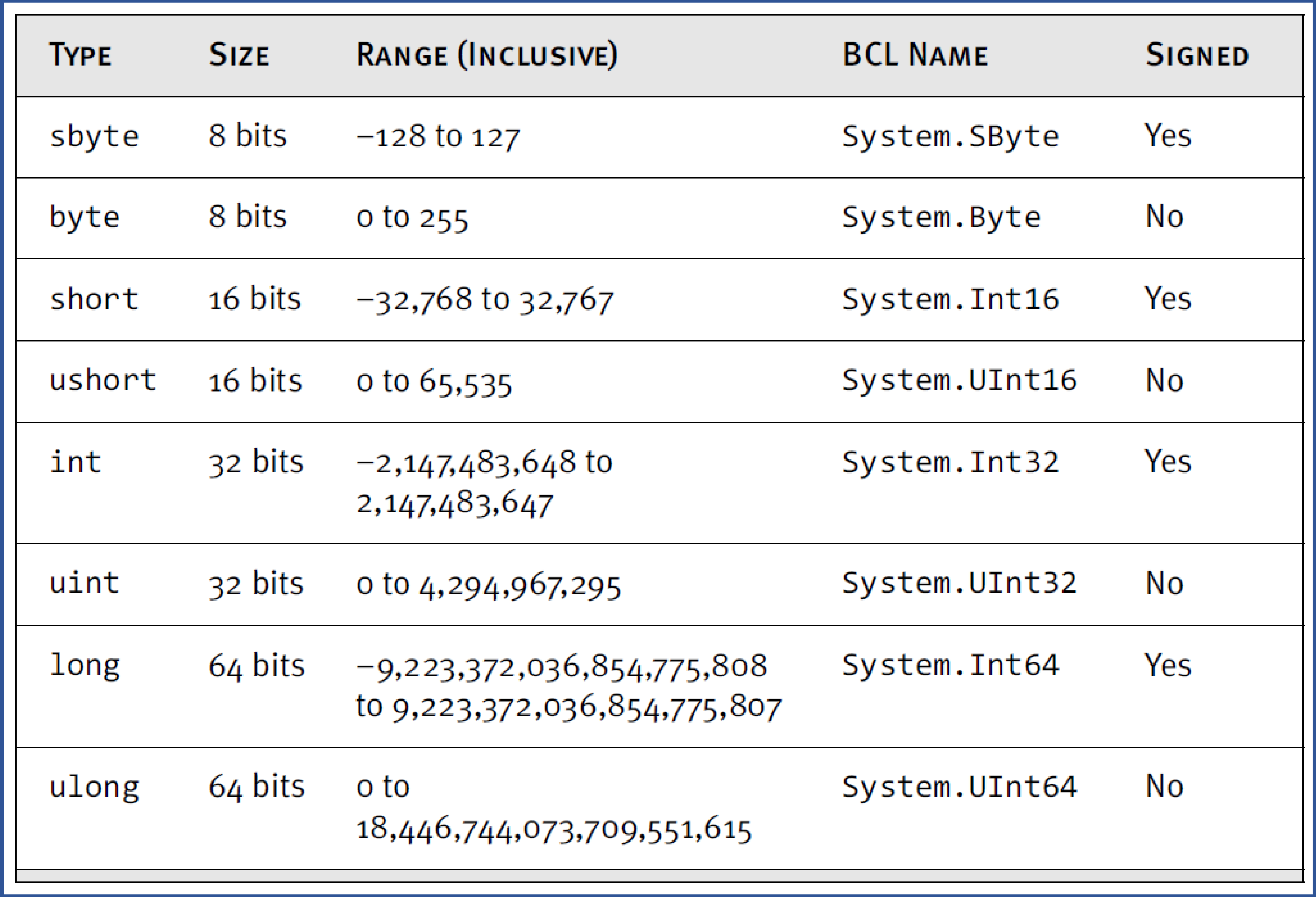


protected	public	readonly	ref
remove*	return	sbyte	sealed
select*	set*	short	sizeof
stackalloc	static	string	struct
switch	this	throw	true
try	typeof	uint	ulong
unchecked	unsafe	ushort	using
value*	virtual	void	volatile
where*	while	yield*	

* Contextual keyword

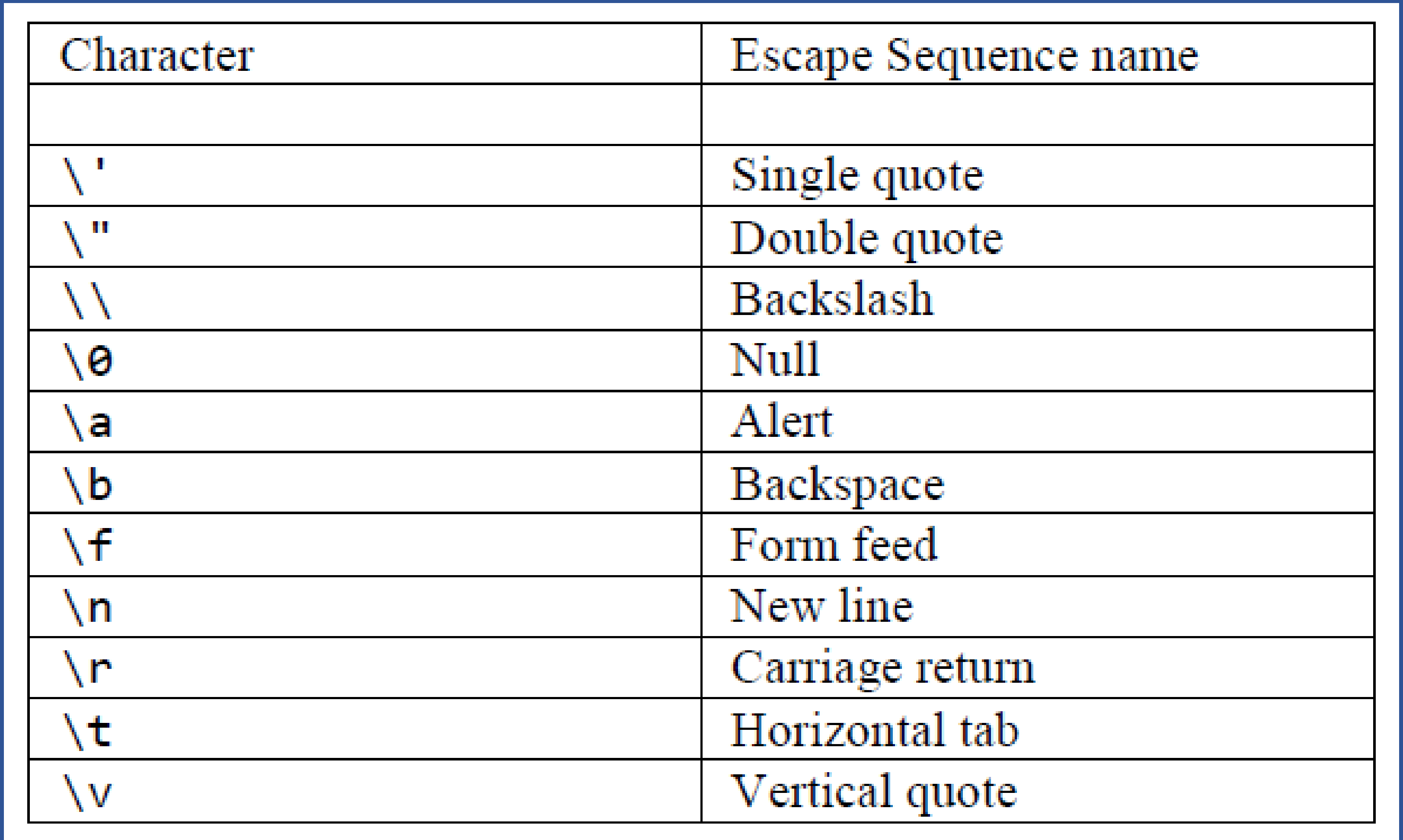
Data Types


Data Types
1 Numeric Types
Integer Types
Floating-Point Types
Decimal Type
Literal Values
2 More Types
Boolean Type
Character Type
Strings
3 null and void
4 Categories of Types
Value Types
Reference Types
5 Conversions
Explicit Cast
Implicit Cast
Without Casting
6 Arrays
Declaring
Instantiating
Assigning
Using
Strings as Arrays



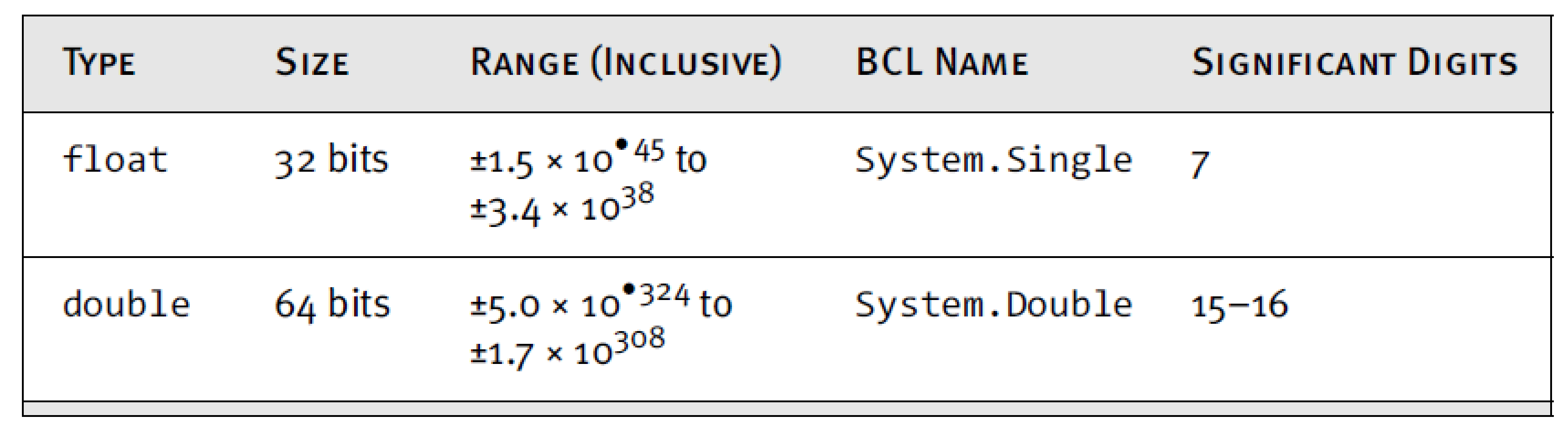
TYPE	SIZE	RANGE (INCLUSIVE)	BCL NAME	SIGNED
sbyte	8 bits	−128 to 127	System.SByte	Yes
byte	8 bits	0 to 255	System.Byte	No
short	16 bits	−32,768 to 32,767	System.Int16	Yes
ushort	16 bits	0 to 65,535	System.UInt16	No
int	32 bits	−2,147,483,648 to 2,147,483,647	System.Int32	Yes
uint	32 bits	0 to 4,294,967,295	System.UInt32	No
long	64 bits	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	System.Int64	Yes
ulong	64 bits	0 to 18,446,744,073,709,551,615	System.UInt64	No

Character Escape Sequences



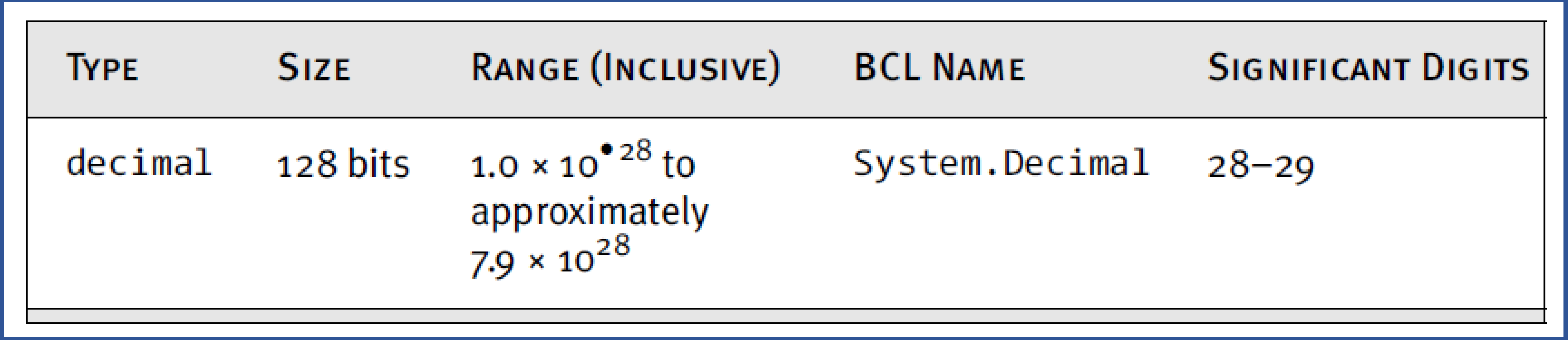
Character	Escape Sequence name
`\'`	Single quote
`\"`	Double quote
`\\`	Backslash
`\0`	Null
`\a`	Alert
`\b`	Backspace
`\f`	Form feed
`\n`	New line
`\r`	Carriage return
`\t`	Horizontal tab
`\v`	Vertical quote

Floating-point



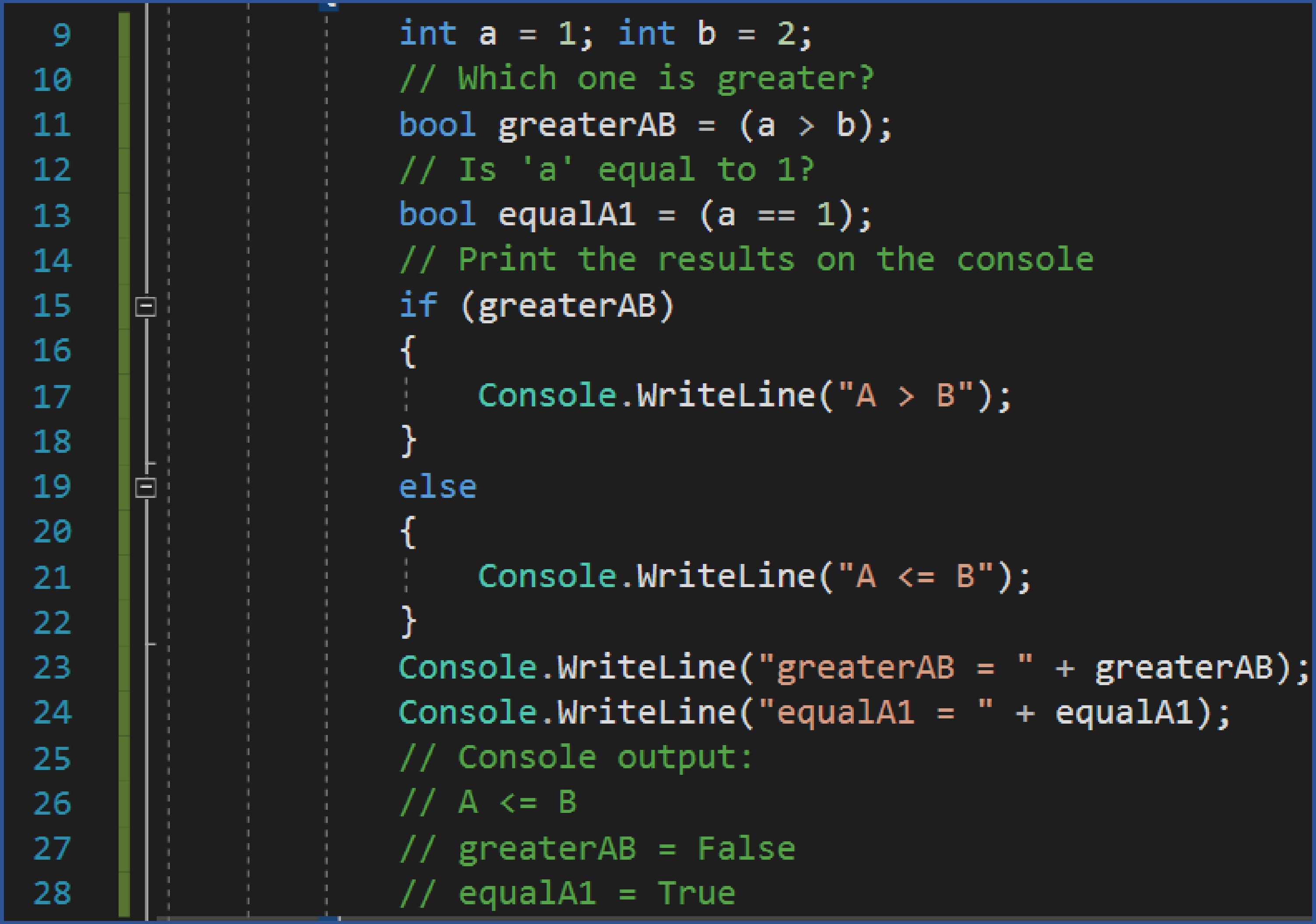
Type	Size	Range (Inclusive)	BCL Name	Significant Digits
float	32 bits	$\pm 1.5 \times 10^{\bullet 45}$ to $\pm 3.4 \times 10^{38}$	System.Single	7
double	64 bits	$\pm 5.0 \times 10^{\bullet 324}$ to $\pm 1.7 \times 10^{308}$	System.Double	15–16

Decimal



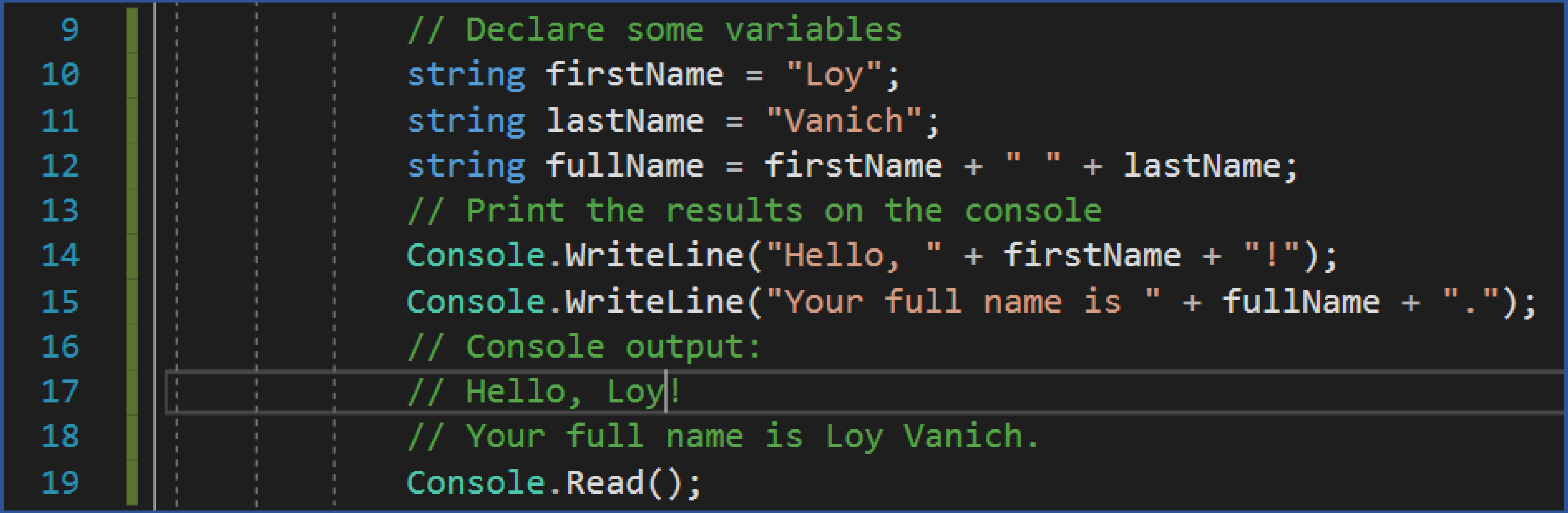
Type	Size	Range (Inclusive)	BCL Name	Significant Digits
decimal	128 bits	$1.0 \times 10^{\bullet 28}$ to approximately 7.9×10^{28}	System.Decimal	28–29

Boolean Type



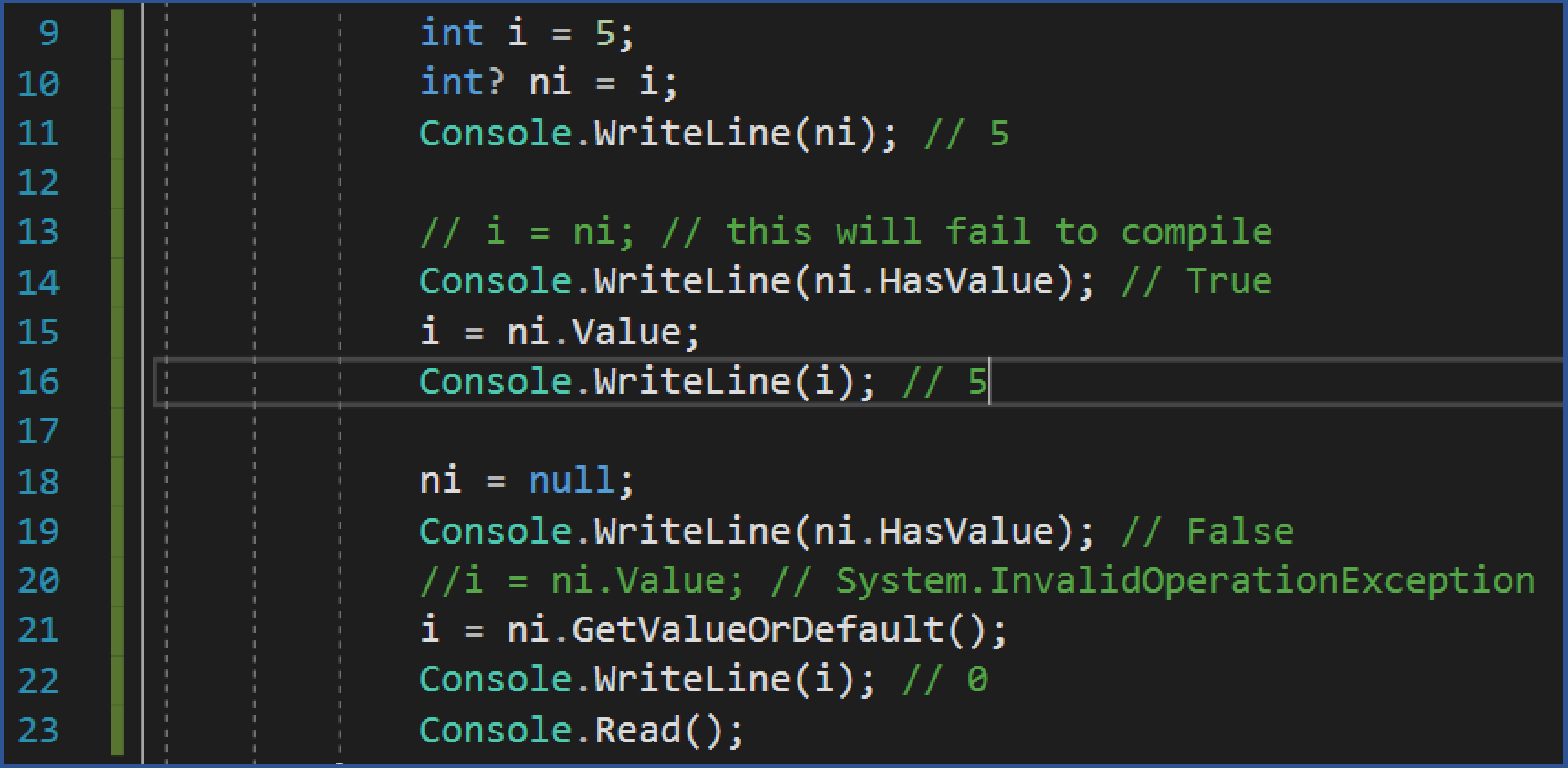
```
int a = 1; int b = 2;
// Which one is greater?
bool greaterAB = (a > b);
// Is 'a' equal to 1?
bool equalA1 = (a == 1);
// Print the results on the console
if (greaterAB)
{
    Console.WriteLine("A > B");
}
else
{
    Console.WriteLine("A <= B");
}
Console.WriteLine("greaterAB = " + greaterAB);
Console.WriteLine("equalA1 = " + equalA1);
// Console output:
// A <= B
// greaterAB = False
// equalA1 = True
```

Strings



```csharp
            // Declare some variables
            string firstName = "Loy";
            string lastName = "Vanich";
            string fullName = firstName + " " + lastName;
            // Print the results on the console
            Console.WriteLine("Hello, " + firstName + "!");
            Console.WriteLine("Your full name is " + fullName + ".");
            // Console output:
            // Hello, Loy!
            // Your full name is Loy Vanich.
            Console.Read();
```

Nullable Types

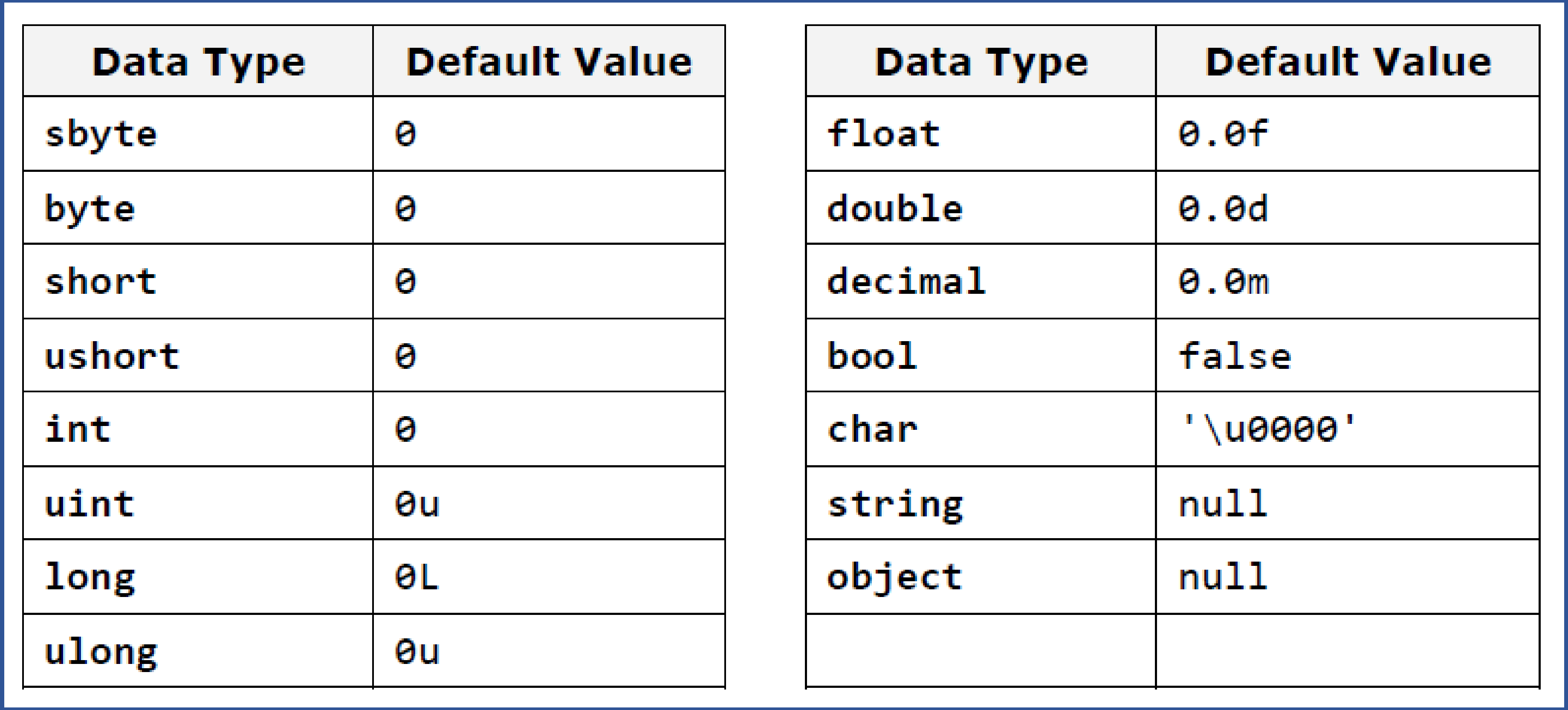


```
            int i = 5;
            int? ni = i;
            Console.WriteLine(ni); // 5

            // i = ni; // this will fail to compile
            Console.WriteLine(ni.HasValue); // True
            i = ni.Value;
            Console.WriteLine(i); // 5

            ni = null;
            Console.WriteLine(ni.HasValue); // False
            //i = ni.Value; // System.InvalidOperationException
            i = ni.GetValueOrDefault();
            Console.WriteLine(i); // 0
            Console.Read();
```

Variable Rules

- Variable names can contain the letters a-z, A-Z, the digits 0-9 as well as the character '_'.
- Variable names cannot start with a digit.
- Variable names cannot coincide with a keyword of the C# language. For example, base, char, default, int, object, this, null and many others cannot be used as variable names.

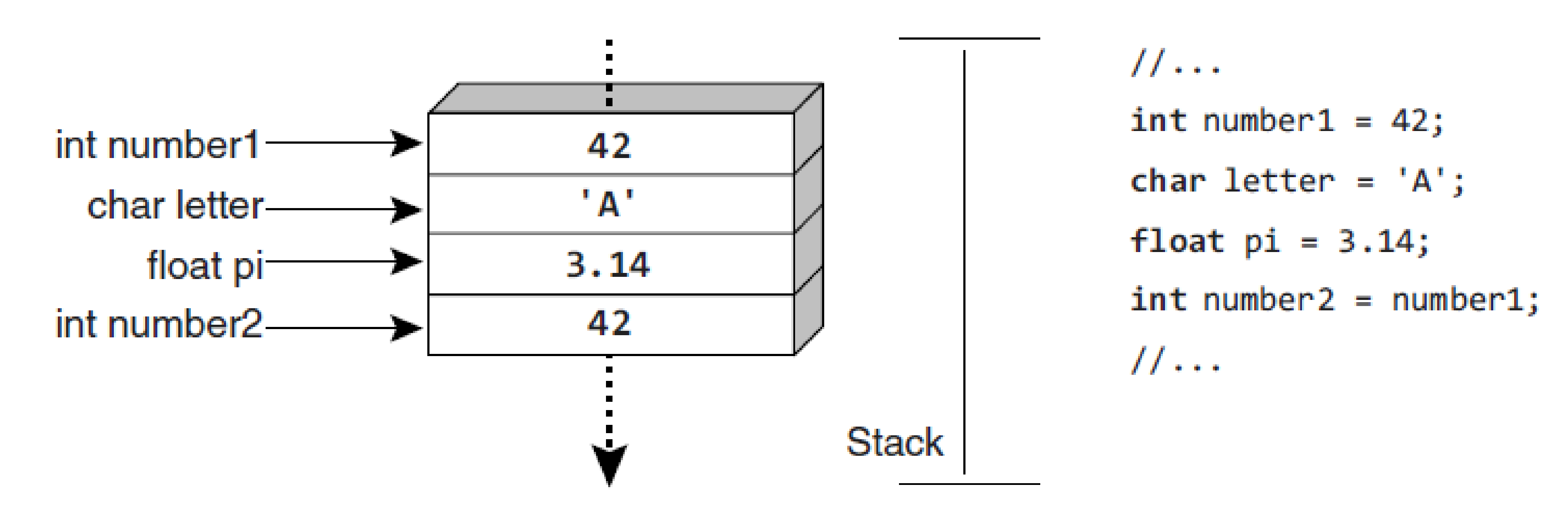
Default Variable Values



Data Type	Default Value
sbyte	0
byte	0
short	0
ushort	0
int	0
uint	0u
long	0L
ulong	0u

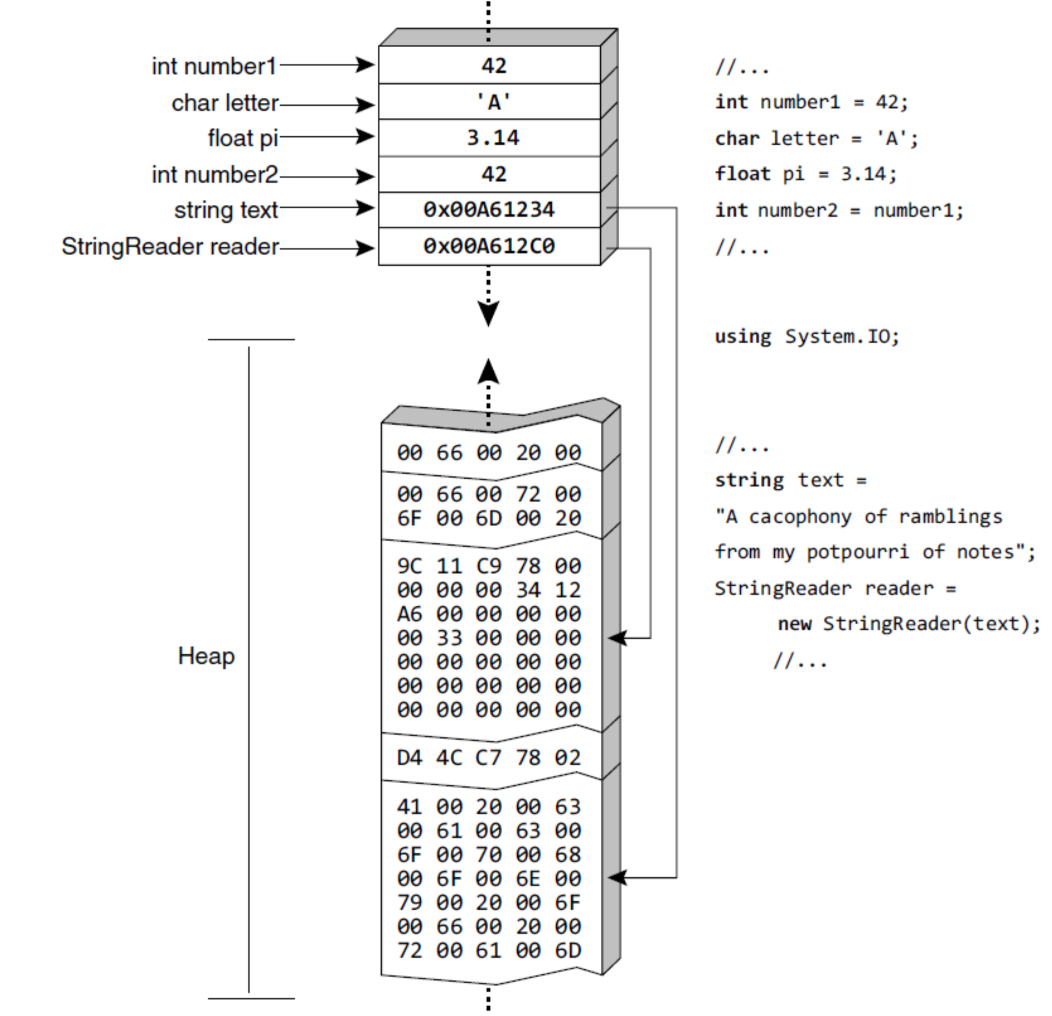
Data Type	Default Value
float	0.0f
double	0.0d
decimal	0.0m
bool	false
char	'\u0000'
string	null
object	null

Value Type



```
//...
int number1 = 42;
char letter = 'A';
float pi = 3.14;
int number2 = number1;
//...
```

Reference Type



```
00 66 00 20 00

00 66 00 72 00
6F 00 6D 00 20

9C 11 C9 78 00
00 00 00 34 12
A6 00 00 00 00
00 33 00 00 00
00 00 00 00 00
00 00 00 00 00
00 00 00 00 00

D4 4C C7 78 02

41 00 20 00 63
00 61 00 63 00
6F 00 70 00 68
00 6F 00 6E 00
79 00 20 00 6F
00 66 00 20 00
72 00 61 00 6D
```

```
//...
int number1 = 42;
char letter = 'A';
float pi = 3.14;
int number2 = number1;
//...


using System.IO;


//...
string text =
"A cacophony of ramblings
from my potpourri of notes";
StringReader reader =
     new StringReader(text);
     //...
```

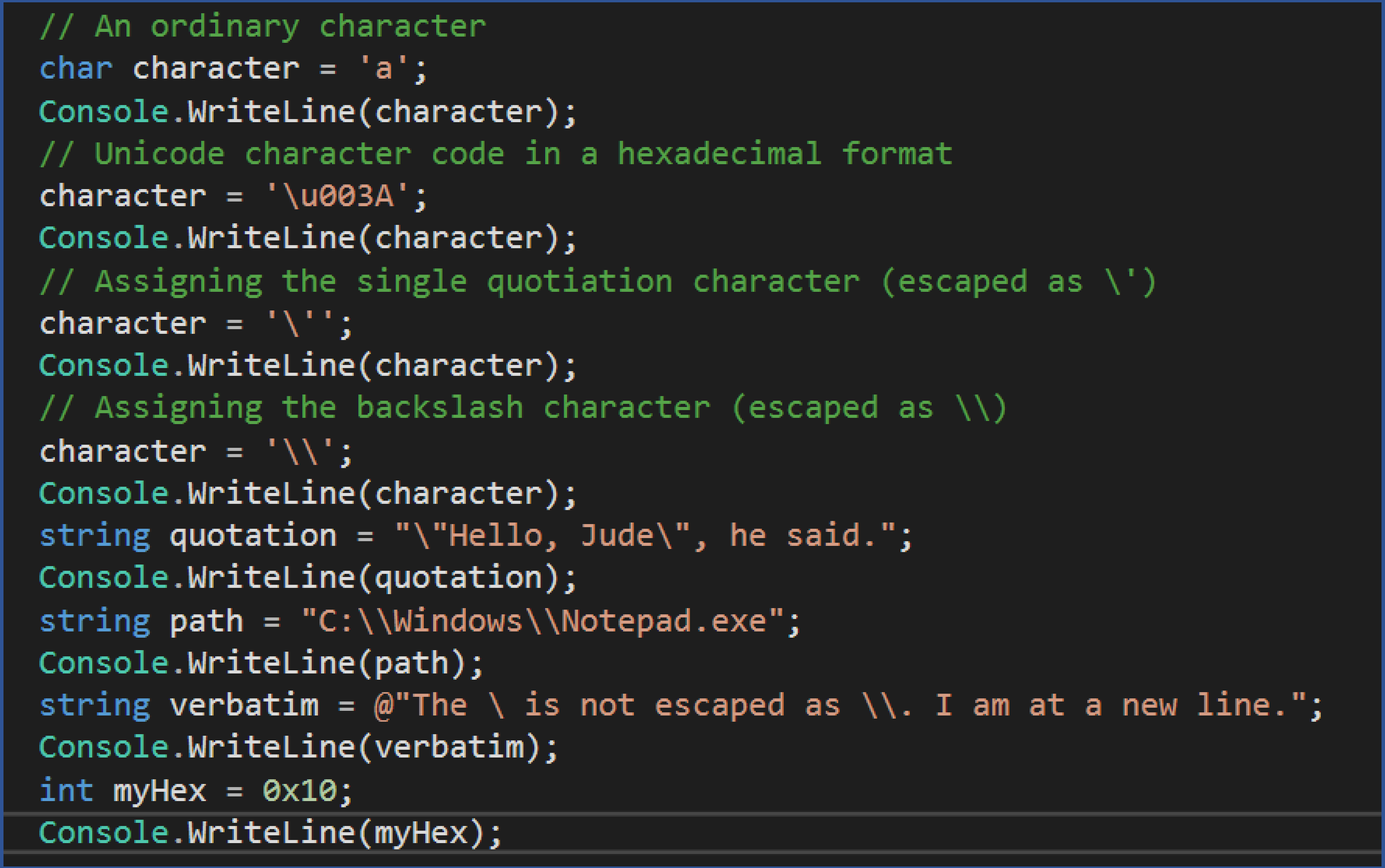
Changing the Type of Data

- Widening and Narrowing
- Casting
- Type conversion
- Number < - > String conversion

Array

- Array declaration
- Array value assignment
- Getting value from array
- Array iteration

Literals



```
// An ordinary character
char character = 'a';
Console.WriteLine(character);
// Unicode character code in a hexadecimal format
character = '\u003A';
Console.WriteLine(character);
// Assigning the single quotiation character (escaped as \')
character = '\'';
Console.WriteLine(character);
// Assigning the backslash character (escaped as \\)
character = '\\';
Console.WriteLine(character);
string quotation = "\"Hello, Jude\", he said.";
Console.WriteLine(quotation);
string path = "C:\\Windows\\Notepad.exe";
Console.WriteLine(path);
string verbatim = @"The \ is not escaped as \\. I am at a new line.";
Console.WriteLine(verbatim);
int myHex = 0x10;
Console.WriteLine(myHex);
```

Exercise

1. Write a program that reads your age from the console and prints your age after 10 years.

2. User Console.Write to Beep 5 times

3. Declare several variables by selecting for each one of them the most appropriate of the types sbyte, byte, short, ushort, int, uint, long and ulong in order to assign them the following values: 52,130; -115; 4825932; 97; -10000; 20000; 224; 970,700,000; 112; -44; -1,000,000; 1990; 123456789123456789.

4. Initialize a variable of type int with a value of 256 in hexadecimal format (256 is 100 in a numeral system with base 16).

5. Declare a variable of type char and assign it as a value the character, which has Unicode code, 72 (use the Windows calculator in order to find hexadecimal representation of 72).

6. Declare a variable isMale of type bool and assign a value to it depending on your gender.

7. Declare two variables of type string with values "Hello" and "World". Declare a variable of type object. Assign the value obtained of concatenation of the two string variables (add space if necessary) to this variable. Print the variable of type object.

8. Declare two variables of type string and give them values "Hello" and "World". Assign the value obtained by the concatenation of the two variables of type string (do not miss the space in the middle) to a variable of type object. Declare a third variable of type string and initialize it with the value of the variable of type object (you should use type casting).

9. Declare two variables of type string and assign them a value “The "use" of quotations causes difficulties.” (without the outer quotes). In one of the variables use quoted string and in the other do not use it.

10. Write a program to print a figure in the shape of a heart and others

11. Declare two variables of type int. Assign to them values 5 and 10 respectively. Exchange (swap) their values and print them.