JASON ALAN SMITH • ANVILABS

NOTE

.NET / C# Data Types: A Table and Discussion

October 25, 2026 • .NET / C# Katas

data-types c-sharp dotnet memory first-principles value-types

Introduction

Computer programs depend on data types for efficient execution within the host computer. Data types are essential for programs to read from and write to memory. Data types also inform programs which operations may be performed on a datum. Given their importance, .NET / C# software engineers must understand the data types available to them.

The Character Type

The *char* type holds a single Unicode character.

Copyright © 2026 Jason Alan Smith

C# Type Name	.NET BCL Type Name	Size	Minimum Value	Maximum Value
char	System.Char	2 Bytes	U+0000	U+FFFF

Copyright © 2026 Jason Alan Smith

A char is a single Unicode UTF-16 code unit. It usually maps to one written character, but some characters need two chars (a surrogate pair).

The Integer Types

Software engineers have ten integer data types built into C# at their disposal.

Copyright © 2026 Jason Alan Smith

C# Type Name	.NET BCL Type Name	Size	Minimum Value	Maximum Value
sbyte	System.SByte	1 Byte	-128	127
byte	System.Byte	1 Byte	0	255
short	System.Int16	2 Bytes	-32,768	32,767
ushort	System.UInt16	2 Bytes	0	65,535
int	System.Int32	4 Bytes	-2,147,483,648	2,147,483,647
uint	System.UInt32	4 Bytes	0	4,294,967,295
long	System.Int64	8 Bytes	-9,223,372,036,854,775,808	9,223,372,036,854,775,807

Copyright © 2026 Jason Alan Smith

C# Type Name	.NET BCL Type Name	Size	Minimum Value	Maximum Value
ulong	System.UInt64	8 Bytes	0	18,446,744,073,709,551,615
nint	System.IntPtr	4 or 8 Bytes	Depends on platform (computed at runtime)	Depends on platform (computed at runtime)
nuint	System.UIntPtr	4 or 8 Bytes	Depends on platform (computed at runtime)	Depends on platform (computed at runtime)

Copyright © 2026 Jason Alan Smith

A key distinction between these integer data types is that each is a pair of two types. One type in the pair holds a signed value. This means the value can be negative or positive. The other type in the pair holds an unsigned value—the value can only be zero or positive—that is, nonnegative. Types such as sbyte, short, int, and long are signed types. Types such as byte, ushort, uint, and ulong hold unsigned values.

The Floating-Point Types

In addition to the ten integer data types built into C#, software engineers have three floating-point data types available to them.

Copyright © 2026 Jason Alan Smith

C# Type Name	.NET BCL Type Name	Precision	Size	Minimum Magnitude	Maximum Magnitude
float	System.Single	~6-9 digits	4 Bytes	$\pm 1.5 \times 10^{-45}$	$\pm 3.4 \times 10^{38}$
double	System.Double	~15-17 digits	8 Bytes	$\pm 5.0 \times 10^{-324}$	$\pm 1.7 \times 10^{308}$
decimal	System.Decimal	28-29 digits	16 Bytes	$\pm 1.0 \times 10^{-28}$	$\pm 7.9228 \times 10^{28}$

Copyright © 2026 Jason Alan Smith

C# treats a floating-point literal as a double by default.

It's critical that software engineers realize that the types float and double must *never* be used to store *monetary* values! Rather, the *decimal* type is safe and, in fact, is purpose-built to store such values.

Floating-point numbers exhibit many more intricacies that will be discussed in a future Note.

Numeric Literals

We've already mentioned that C# treats any floating-point number as type double by default. Floating-point types are not the only types that have default values. Integer types have their own defaults.

This raises the question: how do software engineers specify that a floating-point or integer value should be a type other than the default type? The answer is that software engineers depend on *numeric literal suffixes*.

A numeric literal suffix is simply a single letter appended to a literal numeric value. This suffix informs the C# compiler that this literal number should be treated as a type other than its default type.

Copyright © 2026 Jason Alan Smith

Suffix	Alternative Suffix	Data Type
U	u	uint or ulong
L	l	long or ulong
F	f	float
D	d	double
M	m	decimal

Copyright © 2026 Jason Alan Smith

The Boolean Type

Single Unicode characters and numbers are not the only types of data software engineers need to store in memory during program execution. Many logical comparisons or conditions occur during program execution. The Boolean data type is purpose-built for precisely such logical operations.

Copyright © 2026 Jason Alan Smith

C# Type Name	.NET BCL Type Name	Size	False Value	True Value
bool	System.Boolean	1 Byte	false	true

Copyright © 2026 Jason Alan Smith

Summary

.NET / C# software engineers have a variety of data types at their disposal. They include ten integer types, one Unicode character type, three floating-point types, and one Boolean type. Software engineers use suffixes to inform the C# compiler that numeric literals should be a type other than their default type. A crucial takeaway: for any monetary value, use the decimal data type!

Copyright © 2026 Jason Alan Smith