



Chapter 3. Introduces the C# language

A reference of MSDN Library for Visual Studio 2017

IT Faculty, TDM University

Contents

- Introduction
- Types
- Conditional Branching Statements
- Iteration Statements
- Mathematical Operators

Introduction

- The goal of C# is to provide a simple, safe, object-oriented, highperformance language for .NET development.
- The C# language is disarmingly simple, with only about 80 keywords
- C# includes all the support for structured, component-based, object-oriented programming that one expects of a modern language built on the shoulders of C++ and Java.

Example

- ***Example: A simple "Hello World" program in C#***

```
class HelloWorld
```

```
{
```

```
static void Main( )
```

```
{
```

```
    // Use the system console object
```

```
    System.Console.WriteLine("Hello World");
```

```
}
```

```
}
```

- Compiling and running HelloWorld displays the words "Hello World" at the console. Let's take a closer look at this simple program.

Introduction

- **Methods**
 - A *method* is a *function* owned by your class.
- **Comments**
 - C# supports three types of comments.
 - `//` comments
 - `/*` comments `*/`
 - `///`
- **Console Applications**
- **Namespaces**
- **The using Keyword**
- **Case Sensitivity**
 - C# is case-sensitive, which means that `writeLine` is not the same as `WriteLine`

Types

Table 3-1. C# built-in value types

Type	Size (in bytes)	.NET Type	Description
byte	1	Byte	Unsigned (values 0-255).
char	1	Char	Unicode characters.
bool	1	Boolean	true or false.
sbyte	1	Sbyte	Signed (values -128 to 127).
short	2	Int16	Signed (short) (values -32,768 to 32,767).
ushort	2	UInt16	Unsigned (short) (values 0 to 65,535).
int	4	Int32	Signed integer values between -2,147,483,647 and 2,147,483,647.
uint	4	UInt32	Unsigned integer values between 0 and 4,294,967,295.
float	4	Single	Floating point number. Holds the values from approximately $\pm 1.5 \times 10^{-45}$ to approximate $\pm 3.4 \times 10^{38}$ with 7 significant figures.
double	8	Double	Double-precision floating point; holds the values from approximately $\pm 5.0 \times 10^{-324}$ to approximate $\pm 1.7 \times 10^{308}$ with 15-16 significant figures.
decimal	8	Decimal	Fixed-precision up to 28 digits and the position of the decimal point. This is typically used in financial calculations. Requires the suffix "m" or "M."
long	8	Int64	Signed integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
ulong	8	UInt64	Unsigned integers ranging from 0 to 0xffffffffffffffff.

Variable

- Declared
type identify [=value];
- Example
int i = 0;
int y,x,z;
float a = 1;
string st = "";

Types

■ Converting built-in types

- `short x = 5;`
- `int y = x; // implicit conversion`

- `short x;`
- `int y = 500;`
- `x = y; // won't compile`

- `short x;`
- `int y = 500;`
- `x = (short) y; // OK`

Conditional Branching Statements

- If...else statements
 - if** (expression)
Statement1;
 - [else**
Statement2;]
- Nested if statements

Conditional Branching Statements

- The switch statement

```
switch (expression)
{
    case constant-expression:
        statement
        jump-statement
    [default: statement]
}
```

Iteration Statements

- **The while loop**
while (expression) statement;
- **do . . . while loop**
do expression while statement;
- **The for loop**
for ([initializers]; [expression]; [iterators]) statement;
- **The foreach statement**
foreach (<set type> <object> in < set name>)
statement;

Mathematical Operators

- Simple arithmetical operators (+, -, *, /)
- The modulus operator (%) to return remainders
- Increment and Decrement Operators
 - +=
 - -=
- The prefix and postfix operators
 - ++
 - --
- Use of Logical Operators with Conditionals
- The Ternary Operator
 - `cond-expr ? expr1 : expr2`