**5) c# language fundamentals.**

*C# and .NET Programming* demonstrates a very simple C# program that prints the text string "Hello World!" to the console screen and provides a line-by-line analysis of that program. However, even that simple program was complex enough that we had to skip some of the details. In this chapter, we'll begin an in-depth exploration of the syntax and structure of the C# language. The *syntax* of a language is the order of the keywords, where you put semicolons, and so forth. The *semantics* are what you are expressing in the code, and how your code fits together.

.

Statements

In C#, a complete program instruction is called a *statement* and each statement ends with a semicolon (;). Forgetting a semicolon is a very common mistake for novice programmers, but Visual Studio will catch you if you do it. Programs consist of sequences of statements, such as the following:

int my Variable; // a statement

my Variable = 23; // another statement

int another Variable = my Variable; // yet another statement

2.1Type

C# is a *strongly typed* language. That means that every object you create or use in a C# program must have a specific *type*. In other words, you must declare the object to be an integer or a string or a Dog or a Button. Essentially, the type indicates the characteristics of the object and what it can do.

**Table 3.1. The intrinsic types built into C#**

|  |  |  |  |
| --- | --- | --- | --- |
| **C# type** | **Size (in bytes)** | **.NET type** | **Description** |
| Byte | 1 | Byte | Unsigned (values between 0 and 255). |
| Char | 2 | Char | Unicode characters (a modern way of storing most characters, including international language characters). |
| Bool | 1 | Boolean | True or false. |
| S byte | 1 | S Byte | Signed (values between -128 and 127). |
| Short | 2 | Int16 | Signed (short) (values between -32,768 and 32,767). |
| U short | 2 | UInt16 | Unsigned (short) (values between 0 and 65,535). |
| Int | 4 | Int32 | Signed integer values between -2,147,483,648 and 2,147,483,647. |
| Unit | 4 | UInt32 | Unsigned integer values between 0 and 4,294,967,295. |
| Float | 4 | Single | Floating-point number. Holds the values from approximately +/-1.5 x 10-45 to approximately +/-3.4 x 1038 with seven significant figures. |
| Double | 8 | Double | Double-precision floating-point. Holds the values from approximately +/-5.0 x 10-324 to approximately +/-1.8 x 10308 with 15 to 16 significant figures. |
| Decimal | 12 | Decimal | Fixed-precision up to 28 digits and the position of the decimal point. This type is typically used in financial calculations. Requires the suffix "m" or "M" when you declare a constant. |
| Long | 8 | Int64 | Signed integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. |
| Long | 8 | UInt64 | Unsigned integers ranging from 0 to approximately 1.85 x 1019. |
|  |  |  |  |