**C# and the most common languages**

**C#** features:

* one of the newest programming languages;
* will become a very popular language for a number of reasons - one of the key reasons is Microsoft and the promises of .NET;
* Visual Basic is really no easier than programming C#. In fact, you can actually write many programs with less code using C#;
* C# was meant to be an advance over both C++ and Java as a general purpose programming language;
* C# clearly includes some constructs that move it beyond its predecessors;
* Some of its features will surely be adopted by programming languages of the near-term future;
* C# removes some of the features of C++ that cause programmers a lot of grief, but no power or functionality was really lost;
* Some of the programming errors that are easy to create in C++ can be totally avoided in C# - this can save you hours or even days in finishing your programs;

**Java**

C# and Java are actually quite similar, from an application developer's perspective.

* Reference types are very similar to pointers in C++, particularly when setting an identifier to some new class instance. But when accessing the properties or methods of this reference type, use the "." operator, which is similar to accessing data instances in C++ that are created on the stack. All class instances are created on the heap by using the new operator, but delete is not allowed, as both languages use their own garbage collection schemes.

**Java**, like C++ and C#, is based on C. If you decide to learn Java later, you will find that a lot of what you learn about C# can be applied.

**C/C++**

C# is directly related to C and C++. C is a root for C++ and C++ is a superset of C. C and C++ shares several syntax, library and functionality. In addition structures, unions, arrays, strings and pointers are most important and similar functionality for both languages. C# inherits most of its operators, keywords, and statements directly from C++. Enums are clearly a meaningful concept in C++. Finally I can clearly say that C# is **the first component-oriented language** in the C/C++ family. C# constructors are very similar with C++ constructors. Like C++, methods are non-virtual by default, but can be marked as virtual. There is also some difference between C# and C++, C# supports multiple inheritance of interfaces, but not of classes. Another difference is destructors, their syntax is same with C++ but actually they are very different.

Most of the C# basic types have the same names as C++ basic types but they are not really same. For example a char in C# is equivalent to a wchar\_t in C++. If you decide to move from C++ to C# there are a few things to watch out to include the changes to new, structs, constructors, and destructors. Creating and using a component (DLL) in C# is fairly easier than in C++. One more thing, Borland’s C++ Builder was a pure C++ with simple RAD environment of Delphi and VB. C++ Builder was not a new language. This is one of the biggest differences between C++ Builder and C#. The CLR (Common Language Runtime) improves runtime interactivity between program development simplicity, security and portability. However CLR gives usability for cross-language integration. In addition to all those CLL has+ a perfect foundation for a rich set of class libraries.