Improve Performance with Native C/C++

In this chapter, you will learn about:

- The benefits and costs to integrating C/C++ code into your iPhone application.
- Basic concepts in C programming
  - Data types
  - Pointer
  - Memory management
- Basic concepts in C++ programming
  - Class
  - Memory management
  - Inheritance
  - Template
- How to work through a real example with SQLite, a database that has a C API.
- How to integrate C++ and Objective-C++ into your iPhone application.

In this chapter, you will learn about low-level programming with C/C++, which is important for high performance applications. It’s true that Objective-C is a superset of C and also a native programming language, but Objective-C adds a wrapper over the C language that reduces performance. If you have ever worked with games and animations, you know that using OpenGL with C/C++ provides much better performance.
C++ is also supported by Apple. Most of your basic applications will not need to touch any C/C++ code; however, when your application requires better performance, you should consider this option. Moreover, you don’t need to write lots of C/C++ code but you do need to understand C/C++ code to call the library correctly. You may also need to modify open source libraries to fit your needs.

Because Objective-C is a superset of C, everything you can do in C, you can do in Objective-C. There are syntax differences and new concepts, but I’ll cover them in this chapter. C++ and Objective-C have many differing concepts as well, so it may be harder for you to learn C++ than C.

Actually, the concept of Objective-C is limited, and most of the classes and supports you see in the iPhone development environment come from Cocoa Touch. For simplicity, I will use Objective-C as a symbol for Objective-C and all the supports from Cocoa Touch Frameworks.

**Benefits and Costs**

Before going into the ideas of C/C++ programming, I want to quickly analyze the benefits and costs of using C/C++ code inside your iOS application.

**Benefits:**

- There are specialized libraries written in C/C++ such as animation or sound libraries. These libraries are usually written in C/C++ for high performance and portability.
- Your application can be ported to Android without much effort.
- You may improve application performance by using C/C++ code.

**Cost:**

- C/C++ has a different syntax than Objective-C, and mixing them up makes the code harder to understand.
- C/C++ has a different memory management mechanism than Objective-C, so you need to be careful about memory leaks or application crashes.

So, knowing both benefits and costs, you can make the decision if you want to integrate C/C++ into your iPhone application or not. Even if you are using another open source library or write your own code, you should understand C/C++ before doing so. Many of the problems that arise can be so subtle that iPhone instruments can’t help you much. It’s easy to spend an hour integrating a library—and then spend the whole day fixing a bug inside.

Objective-C is already a native programming language, so you may not get much better performance if you try to write all code in C/C++. However, there are many high performance libraries written in C/C++ that you can take advantage of and integrate into your application.