Who Should Read This Book

This book was written for anyone who wants to understand how C++ programming works. *Using* C++ will get you up to speed fast on the essential elements of C++ that you will use every day in your programs. It provides you with a thorough grounding in using the power of the C++ language to create object-oriented programs. Whether you already know how to program in some other language or have never programmed before, this book will help you learn C++ quickly and will help to simplify the often complex application development process.

What Experience Do You Need?

The concepts and sample code presented in this book should work under any standard C++ compiler, with few exceptions. If you're familiar and comfortable with fundamental programming and object-oriented concepts, then you'll have a head start—but the book assumes no previous programming experience in C, C++, or any other language.

The History and Evolution of C++

The C++ language was conceived and developed in the early 1980s by Dr. Bjarne Stroustrup at AT&T's Bell Labs in New Jersey. Dr. Stroustrup developed C++ as an object-oriented C, or "C with Classes" as it was originally called. The C programming language was developed at Bell Labs a decade earlier, and by the early 1980s, C had become the *lingua franca* of computer programming.

Dr. Stroustrup wanted an elegant, object-oriented programming language that would be compatible with existing C code; C++ was the result. Even today, after several developmental and conceptual changes, C++ retains full compatibility with C. The C programming language was finally standardized in 1989 by a joint committee made up of representatives from the American National Standards Institute (ANSI) and the International Standards Organization (ISO).

Just after C was standardized, another joint committee was formed to look into standardizing C++. After seven years of modification and purposeful evolution of C++ into the industrial strength, robust programming language it is today, the standards committee approved a final draft of the C++ standard, and Standard C++ became a reality at long last in February 1998.

Today, C++ is quickly replacing C as the predominant programming language on the planet. In terms of flexibility, speed, portability, and power, C++ has no equal.

How This Book Is Organized

The book is divided into six parts, each part covering a specific set of related concepts or issues. Before you dive in and start reading about how to write programs with C++, please take the time to peruse this brief overview.

Part I: C++ Basics

In this first part of the book, you learn the basic syntax and idiosyncrasies of the C++ language. Topics covered in Part I include data types, expressions and statements, programmatic decision making, loops, arrays, and functions.

Part II: Beyond the Basics

In the second part of this book, you'll learn about some more advanced features of the C++ language. Topics covered in Part II include function and operator overloading, user-defined data types (structures and unions), pointers and references, dynamic memory allocation, preprocessing, function pointers, error trapping, namespaces, bit manipulation, and the program build process.

Part III: Object-Oriented Programming

In the third part of this book, you are introduced to object-oriented programming and the C++ language features that support this programming model. Topics covered in Part III include C++ classes, class inheritance and polymorphism, dynamic storage, object-oriented design and development, class templates, dynamic casting, and runtime type identification.

Part IV: Using the Standard C++ Library

In the fourth part of this book, you'll discover the advantages and details of using the Standard C++ Library, an international C++ standard set of functions, classes, and algorithms. Topics covered in Part IV include iterators and containers, standard strings, numerics, files and streams, and standard algorithms.