Over 150,0

TEACH YOURSELF

"Herb Schildt tells his programmers what they want and need to know- simply, clearly, concisely, and authoritatively."

-ACM Computing Reviews

C

BONUS
http://www.osborne.com
Free Code on Web Page

Third Edition

The Most Successful and Proven Method of Learning C

- Master all C fundamentals •
- Build your skills with hundreds of examples and exercises •
- Take charge of advanced features and techniques •

Herbert Schildt

Best-Selling C/C++ Author with More Than 1.5 Million Books Sold

Teach yourself C

Third Edition

Herbert Schildt

DITION-2003-2004

Osborne McGraw-Hill
Berkeley New York St Louis San Francisco Auckland Bogota Hamburg London ..land Mexico City
Milan Montreal New Delhi Panania City Paris Sao Paulo Singapore S,uney Tokyo Toronto

Publisher
Brandon A. Nordin

Editor-In-Chief
Scott Rogers

Acquisitions Editor
Wendy Rinaldi

Project Editor
Heidi Poulin
Ann Sellers

Technical Editor

Gregg Guntle

Copy Editor

Erik Van Eaton

Proofreader

Karen Mead

Indexer

Sheryl Schildt
Computer Designers

Roberta Steele
Sylvia Brown

Illustrtor
Roberta Steele

Series Design
Peter Hancik

Osborne **McGraw-HILL**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations or book distributors outsid
U.S.A. or to arrange bulk purchase discounts for sales
Promotions, premiums, or fundraisers, pliease contact
Osborne MCGraw-Hill at the above address.

4567890 DOC 99

ISBN 0-07-882311-0

About the Author

Herbert Schildt is the world's leading programmig author. He is an authority on the C and C++ Languages, a master Windows programmer, and an expert on Java. His programming books have sold early two million copies worldwide and have been translated Ito all major foreign languages. He is the author of numerous best-sellers, including c: The Complete Reference, C++: The Complete Reference, C++ from the Ground Up, FMC programming from the Ground Up, Windows 95 Programming in C and C++, Windows NT 4 Programming form the Ground Up, and many others. He is also a member of both the ANSI C and C++ standardization committees.

Contents

Preface

This book teaches you how to program in what is usually regarded as the world.s most important professional programming language: C.

One reason for C's success and staying power is that programmers like it. C combines subtlety and elegance with raw power and flexibility. It is a structured language that does not confine. It is a high-performance language that does not constrain. C is also a language that puts you, the programmer, firmly in charge. C was creatted by a programmeer for programmers. It is not the contrived product of a committee, but rather the outco... of programmers seeking a better programming language.

C is important for another reason. It is the gateway to the world's two other professional programming languages: C++ and java. C++ is built upon C, and Java is built upon C++. Thus, C is at the foundation of all modern programming, and knowledge of C is fundamental to the successful creation of high-performance, high-quality software. Simply put, to be a professional programmer today means that you are. competent in C.

A Short History of C

C was invented and first implemented by Dennis ritchie oon a DEC PDP-11 using the UNIX operating system. C is the result of a development process that started with an older language called BCPL, developed by Martin Richards. BCPL influenced a language called B that was invented by ken Thompson and that led to the development of C in the 1970s.

For many years, the de facto standard for C was the one described in The C programming Language by Brian Kernighan and Dennis Ritchie (Prentice-Hall,) 1978. However, as C grew in popularity, a committee was oranizd in 1983 to create an ANSI (American National Standards Institute) standard for C. The standardization process took six years (much longer than anyone reasonably expected). The ANSI C standard was finally adopted late in 1989 and

TEACH YOURSELF
C

The first copies become generally available in 1990. The star...ard was amended slightly in 1996. Today, virtually all C compilers comply with ANSI standard C ad that is the version of C you will learn in this book. (That is, this book teaches ANSI standard C.)

C is often referred to as a middle-language. Before C there were basically two types of languages used to program computers. One is called assembly language, which is the symbolic representation of the actual machine instructions executed by the computer. Assembly language is a low lavel language because the programmer is working with (in s...bolic form) the actual instructions that the computer wi.i execute. Assembly language can be used to create very ...ficient programs, but it provides no built-in control structures or I O funct..ns. All such items must be manually constructed by the programmer. By contrast, a high-level language buffers the programmer from the computer. A high-level language typically supplies various control structures, input and output commands, and the like, which make programming easier and faster. However, the elements of a high-level ..anguage may not relate directly to the way that the computer ultimately carri.. out the program. This separation often causes programs written using a high-level language to be less efficient than those written in assembly language. Because many people find assembly language programming to be a tedious, difficult task, there was a need for a language that balanced ease-ofuse with efficiency. Many programmers feel that C provides this balance. It successfully combines the structure of a high-level language with the power and efficiency of assembly language. Since it spans the up between assembly language and High-level languages, it is called a middle-level language.

Initially, C was used primarily for creating systems software. Systems software consists of th..se programs that help run the compater. These include programs such as operating systems, compilers, and editors. However, as C gained in popularity, it began to be used for general purpose programming. Today, C is used by programmers for virtually any programming task. It is a language that has survived the test of time and proven itself to be as versatile as it is powerful.

C vs. C++

Newcomers are sometimes confused about the differences between C and C++ and how they relate to each other. In short, C++ is an extended version of C that is designed to support object-oriented programming (jOOP). C++ contains and supports the entire C language in addition to a set of object-oriented extensions. (That i. C++ is a superset of c.) Because C++ is built upo the foundatio of C. you cannot learn C++ without learning the basics of C. Therefore, if you think that you will someday move o to C++ , your knowledge of C will not only be useful, it will be necessary.

About this book

This book is unique because it teaches you the C language by aplying mastery learning. It does so by presenting one idea at a time, followed by numerous examples and exercises to help you thorughly understant each topic. This approach ensures that you master each topic before moving on.

The material is presented sequentially. Therefore, you should work carefully through each chapter because each chapter assumes that you know the material presented in all preceding chapters.

This book teaches ANSI standard C. This ensures that your knowledge will be applicable to the wieest range of C environents. This book also uses contemporary syntax and structure, which means that you will be learning the right way to write C programs form the very beginning.

Now This Book is Organized

This book is composed of 12 chapters ad 4 appendices. Each chapter (except Chapter 1) begins with a Review Skills Check, which consists of questions and exericises covering the provious chapter's material. The chapters are divided into sections. Each section covers one topic. At the and of each section are examples followed by exercises that test your understanding of the topic. At the end of each chapter, you will find a Mastery Skills check, which checks your knowledge of the material in the chapter. Finally, a Cumulative Skills Check is

TEACH YOURSELF
C

Presented that tests how well you are integrating new material with that presented in earlier chapters.

What's New in the Third Edition

For the most part. the form and structure of this book are unchanged from the previous tow editions. Since C is a stable. standardized language, there was no reason to make major revisions. The two significant changes are the inclusion of full function prototypes in all programs. beginning with Chapter 1. Since all compilers now support-indeed. nearly demand-function prototyping, initial coverage of this issue was moved to the beginning of the book. Another set of changes was prompted by the emerging dominance of 32- bit environments. This caused a number of examples to be rwritten so that they would work for both 16- and 32-bit programs. In some places. additional examples. expanded coverage. or more exercises can be found. Finally. a few changes were made simply to reflect modern coding styles.

Conventions Used in this Book

Whenever a part of a program (such as a variable name) is referenced in text. it will be shown using boldface. Whenever a generic name is referenced in text. it will be shown in italics.

Will a C++ Compller Work with c Programs?

Today, most compilers can compile both C and C++ programs . In fact. it is common to see a compiler advertised as a "C/C++" compiler. or sometimes just as a "C++" compiler. However. any and all compilers that can compile C++ programs can also compile C programs. Therefore. if your compiler calls itself a C++ compiler. don't worry. it is also a full. ANSI-standard C compiler.

If You're Using Windows

If you use Windows and if your goal is to write Windows-based programs, then you have chosen the right language to learn. C is the language for which windows programming was designed, and the language for which Windows programming was designed, and the language in which many Windows programs are written. However, you will not be able to write Windows programs immediately. Here is why.

Windows programs are much more complex to create than non-Windows programs. Even a minimal, do-nothing Windows program maks use of several sophisticated C techniques, such as structures, pointers, and advanced function parameter types. For this reason, it is not possible to teach C programming by writing Windows programs because a Windows program assumes that you are an experienced C programmer!

If you are using Windows (rather than DOS, UNIX, etc.) on your computer, you can still learn C. However, you will need to run your programs from the DOS prompt because they will not be Windows based programs. All modern C/C++ compilers will automatically create the correct environment to execute the programs shown in this Book, so it isn't something that you will typically need to worry about. if you will be writing Windows programs, you will want to read Appendix C after you complete this book. It contains a Windows skeleton program that you can try. Examining this skeleton will also show you that a thorough knowledge of C is required before windows programs can be written.

Don't Forget:

Remember, the source code for all of the programs in this book is available free-of-charge on the Web at http://www.osborne.com. Downloading this code prevents you from having to type in the examples.

For Further Study