Osborne McGraw-Hill

USING MICROSOFT® C

COVERS VERSION 5.1

Werner Feibel

Using Microsoft C

Werner Feibel

Osborne **McGraw-Hill**

Berkeley New York St. Louis San Francisco
Auckland Bogotá Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

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

For information on translations and book distributors outside of the U.S.A., please write to Osborne **McGraw-Hill** at the above address.

A complete list of trademarks appears on page 705.

Using Microsoft C

1234567890

DOCDOC 898 ISBN 0-07-881428-6

Acquisitions Editor	Jeff Pepper
Technical Reviewer	Kris Jamsa
Project Editor	Nancy Beckus
Cover Art	Bay Graphics Design Associates

Screens produced with InSet from InSet Systems, Inc.
Book printed and bound by R.R. Donnelley & Sons Company
Crawfordsville, Indiana

Preface

Welcome to C and to the Microsoft C environment! You've chosen an excellent language and compiler.

About C

C is a small but very powerful programming language. It has always been a popular language among professional programmers, and it is becoming much more widely used as more people are learning to program.

The language has been used to develop all sorts of programs—including simple utilities (such as calendar programs), text editors, database management systems and other applications, as well as operating systems. (Significant portions of such popular operating systems as DOS, OS/2, and UNIX have been written in C.) The C language has even been used to write C compilers!

Because it's a small language, C is quite easy to learn. Because it's a powerful language, you can spend many years learning all of its subtleties and strengths. (Of course, as with any language, you can also spend lots of time searching through frustrating program errors if you're not careful.)

In this book, you'll find a complete introduction to C. You'll learn how to use all the components of the language. You'll also find examples that you can use to begin to explore some of C's finer points. Many of these subtle features are pointed out in the book.

About Microsoft C

Whether you're new to programming or a professional programmer who has been working with other languages, Microsoft C is the ideal programming environment for you. It has everything you'll need to program in C—from your first day at programming through your first professional programming project. It is also the first commercial C compiler to work under OS/2. In fact, your Microsoft C package contains versions for both DOS and OS/2.

The editor in the Microsoft C package is very easy to use. If you're new to computers, this will be helpful. If you generally use a different editor, you may be able to configure the Microsoft editor to obey the same commands as the editor you're already using. You'll find a brief introduction to the editor in Chapter 2.

If you're reluctant to give up the editor you're accustomed to using, you can continue to use this editor—provided it can produce text files (that is, files without any special formatting characters) for the compiler to read.

The Microsoft C compiler is quite a versatile program. If you're just getting started with programming or with C, you can simply let the compiler run in "default" mode—in which the compiler makes decisions concerning various aspects of the program being created. For example, the compiler will try to optimize certain constructs in your program. In this mode, you simply tell the compiler to "compile my file," and the program takes care of the rest. Since there is so little to this process, you'll learn about it in Chapter 1.

By using the compiler in this mode, you can concentrate on designing and writing your programs—confident that the compiler will turn your program design into an executable version, without requiring any special instructions or attention from you.

The Microsoft C package also includes QuickC as a special component. QuickC is a completely integrated programming environment, containing an editor, compiler, and debugger, all in one program.

If you're a professional programmer, you will want to take much greater control of how your program should look. You can set a broad range of options for the compiler—to optimize for speed and program size, as well as to keep track of various settings while building your program.

The programs in this book don't require such fine-tuning. Therefore, the programs are compiled using default settings. You may want to use these programs to explore some of the more advanced options.

The linker is flexible, and is quite easy to use. As with the compiler, you can run the linker in "default" mode, which will suffice for the majority of your programs. If necessary, you can also fine-tune the linker to custom build a program for you.

In addition to these components, the Microsoft C package includes several utilities, such as a program for building precompiled function libraries that you can use in your programs. Once you start writing larger programs, such libraries can help make your programming task much easier. Because they are precompiled, these libraries take less time to process while your program is being built.

The CodeView debugger is also included in your Microsoft C package. This program lets you examine your program as it executes. You can follow the source code as each instruction is carried out. In the debugger, you can also observe the values of specified variables—for example, to determine where an error in your program begins.

Any of these components can be used independently of each other. This gives you considerable leeway in how you create your programs.

When you first begin programming in C, you'll probably just write your programs and compile them. At this point, it's good to know you have new challenges to look forward to. You may even want to explore the other components.

As your programming experience and your challenges grow, you'll begin to use some of the other components to make your job easier and more efficient. For example, you may start building precompiled files and function libraries, for use in other programs. Or you may start using the debugger to examine your program's execution, in order to isolate a bug in your program.

Once you're an experienced programmer, you'll make extensive use of the components, and will have to worry about such issues as program size and version control. Again, it's nice to know that your programming environment allows you to deal with such concerns.

These features make the Microsoft C environment an easy, flexible, and very powerful program collection.

About This Book

In this book you'll learn how to use the C language and your Microsoft C package to accomplish the tasks you want. I've tried to make the learning process as enjoyable as possible. The main thing you'll learn from this book will be how to program well in C.

Where editor commands are discussed, or where specific commands are used to accomplish some task in the program development process, the commands and details will be based on the Microsoft C package. Where C language features are concerned, the discussion is independent of any particular compiler or language implementation. The examples all use the syntax contained in the Draft Proposed ANSI Standard definition for the C language—which will eventually become the "official" definition of the language.

The approach in this book is informal, and uses extensive examples to illustrate C's syntax and features. I strongly urge you to type in the examples and try them. You'll learn C more thoroughly that way.

Many of the examples are designed to enable you to explore a particular feature or issue on your own—either by using or extending the program. In some cases, this goal has led to long examples. Don't let this daunt you; exploration can help you better understand C. If this happens, your gain will more than offset the extra time it took to type in the program.

The book goes into considerable detail about most of C's features. This should make the book useful for both beginners and intermediate students of C. Don't worry if you've never programmed before. There's enough information here to get you started and to keep you going. You'll have to work, however, since the book is no substitute for experience.

If you're already somewhat familiar with C, you may still find useful information here. First, you may find details of C that were not covered in your previous exposure to the language. You may also find topics that may be new to you—for example, complex data structures such as linked lists.

If you're truly interested in learning C, and you're willing to put some effort into it, I think you'll find this book a painless and perhaps even enjoyable way to accomplish your task. When it's fun to do something, you don't mind the extra effort it may require of you. Happy programming!

Acknowledgments

Many people deserve thanks for getting this book out to you.

Kris Jamsa offered valuable criticism and advice in his technical reviews of the book. The book is better as a result of his suggestions, for which I'm grateful.

The people at Osborne/McGraw-Hill deserve special thanks for their patience and perserverance with this project. This book was written during a period of family crises, and everyone at Osborne was very understanding, accommodating, and helpful. Jeff Pepper and Madhu Prasher knew just when to push me and when to wait. Their timing made it easier to write the book, even during the worst of times. Nancy Beckus and the other people in the production department once again worked wonders with the material I sent them over many months. They succeeded admirably, and I congratulate them.

Finally, thanks to the people who have bought my earlier books, since you implicitly encouraged Osborne to let me write this one.

Source Code Listings

There are over 150 programs in this book. If you want to save yourself the task of typing the programs (and about 75 other listings containing functions or header material) into your computer, the source code for all the programs in this book is available on one 720K (3 1/2 inch) or two 360K (5 1/4 inch) diskettes.

The diskettes contain about 450K of source files and cost $22 ($20 + $2 for postage and handling). To order them, please fill out the form on the next page.

Order Form

To order the source code for *Using Microsoft C,* please provide the information requested and include a check or money order for the appropriate amount.

$25 per copy ($20 + $5 for shipping and handling) for foreign orders.
$23 per copy ($20 + $2 for shipping and handling + $1 sales tax) for Massachusetts residents.
$22 per copy ($20 + $2 for shipping and handling) for other orders.

Number of copies: ________ Amount enclosed: $ ________

Diskette format: 360K ____________ 720K ____________

____
Name

____
Company (if applicable)

____
Street Address

____
City State Zip Code

____
Country (if not USA)

Please send this information, along with your check or money order to:

Werner Feibel
P.O. Box 2499
Cambridge, MA 02238-2499

Osborne/McGraw-Hill assumes NO responsibility for this offer. This is solely an offer of Werner Feibel and not of Osborne/McGraw-Hill.

Contents

1 Introducing C

C is an elegantly simple language; it's also a challenge to programmers. At times, you'll love C; at other times, you'll hate it.

There is much to like about C: it's small and powerful. When used properly, C is fast, efficient, and portable, meaning that it's moved easily from one computer or operating system to another.

C lets you do marvelous things, such as building compilers, operating systems, editors, and whatever else you wish. C can also turn nasty, overwriting compilers, operating systems, editors, or whatever else you may have stored in your system's memory.

C gives you the power and potential portability of high-level languages, such as Pascal or Modula-2, along with the flexibility (and destruction potential) of low-level languages, such as assembly language. C lets you build complex data structures found in other high-level languages, and enables you to manipulate individual bits of a data structure in ways normally possible only with assembly language.

However, C will not protect your programs from themselves,