

Introduction

Welcome to *Beginning C: From Novice to Professional, Fourth Edition*. With this book you can become a competent C programmer. In many ways, C is an ideal language with which to learn programming. C is a very compact language, so there isn't a lot of syntax to learn before you can write real applications. In spite of its conciseness and ease, it's also an extremely powerful language that's still widely used by professionals. The power of C is such that it is used for programming at all levels, from device drivers and operating system components to large-scale applications. C compilers are available for virtually every kind of computer, so when you've learned C, you'll be equipped to program in just about any context. Finally, once you know C, you have an excellent base from which you can build an understanding of the object-oriented C++.

My objective in this book is to minimize what I think are the three main hurdles the aspiring programmer must face: coming to grips with the jargon that pervades every programming language, understanding how to *use* the language elements (as opposed to merely knowing what they are), and appreciating how the language is applied in a practical context.

Jargon is an invaluable and virtually indispensable means of communication for the expert professional as well as the competent amateur, so it can't be avoided. My approach is to ensure that you understand the jargon and get comfortable using it in context. In this way, you'll be able to more effectively use the documentation that comes along with most programming products, and also feel comfortable reading and learning from the literature that surrounds most programming languages.

Comprehending the syntax and effects of the language elements is obviously an essential part of learning a language, but appreciating *how* the language features work and *how* they are used is equally important. Rather than just using code fragments, I always provide you with practical working examples that show the relationship of each language feature to specific problems. These examples can then provide a basis for you to experiment and see the effects of changing the code in various ways.

Your understanding of programming in context needs to go beyond the mechanics of applying individual language elements. To help you gain this understanding, I conclude most chapters with a more complex program that applies what you've learned in the chapter. These programs will help you gain the competence and confidence to develop your own applications, and provide you with insight into how you can apply language elements in combination and on a larger scale. Most important, they'll give you an idea of what's involved in designing real programs and managing real code.

It's important to realize a few things that are true for learning any programming language. First, there *is* quite a lot to learn, but this means you'll gain a greater sense of satisfaction when you've mastered it. Second, it's great fun, so you really will enjoy it. Third, you can only learn programming by doing it, and this book helps you along the way. Finally, it's much easier than you think, so you positively *can* do it.

How to Use This Book

Because I believe in the hands-on approach, you'll write your first programs almost immediately. Every chapter has several programs that put a theory into practice, and these examples are key to the book. I advise you to type in and run all the examples that appear in the text because the very act of typing in programs is a tremendous aid to remembering the language elements. You should also attempt all the exercises that appear at the end of each chapter. When you get a program to work for

the first time—particularly when you’re trying to solve your own problems—you’ll find that the great sense of accomplishment and progress make it all worthwhile.

We will start off at a gentle pace, but we’ll gain momentum as we get further into the subject. Each chapter will cover quite a lot of ground, so take your time and make sure you understand everything before moving on. Experimenting with the code and trying out your own ideas is an important part of the learning process. Try modifying the programs and see what else you can make them do—that’s when it gets really interesting. And don’t be afraid to try things out—if you don’t understand how something works, just type in a few variations and see what happens. A good approach is to read each chapter through, get an idea of its scope, and then go back and work through all the examples.

You might find some of the end-of-chapter programs quite difficult. Don’t worry if it’s not all completely clear on the first try. There are bound to be bits that you find difficult to understand at first, because they often apply what you’ve learned to rather complicated problems. And if you really get stuck, you can skip the end-of-chapter programs, move on to the next chapter, and come back to them later. You can even go through the entire book without worrying about them. The point of these programs is that they’re a useful resource for you—even after you’ve finished the book.

Who This Book Is For

Beginning C: From Novice to Professional, Fourth Edition is designed to teach you how to write useful programs as quickly and easily as possible. This is the tutorial for you if

- You’re a newcomer to programming but you want to plunge straight into the C language and learn about programming and writing C programs right from the start.
- You’ve done a little bit of programming before, so you understand the concepts behind it—maybe you’ve used BASIC or PASCAL. Now you’re keen to learn C and develop your programming skills further.

This book doesn’t assume any previous programming knowledge on your part, but it does move quickly from the basics to the real meat of the subject. By the end of *Beginning C*, you’ll have a thorough grounding in programming the C language.

What You Need to Use This Book

To use this book, you’ll need a computer with a C compiler and library installed so that you can execute the examples, and a program text editor for preparing your source code files. The compiler you use should provide good support for the International Standard for the C language, ISO/IEC 9899. You’ll also need an editor for creating and modifying your code. You can use any plain text editor such as Notepad or vi to create your source program files. However, you’ll get along better if your editor is designed for editing C code.

To get the most out of this book you need the willingness to learn, the desire to succeed, and the determination to continue when things are unclear and you can’t see the way ahead. Almost everyone gets a little lost somewhere along the way when learning programming for the first time. When you find you are struggling to grasp some aspect of C, just keep at it—the fog will surely disperse and you’ll wonder why you didn’t understand the topic in the first place. You might believe that doing all this is going to be difficult, but I think you’ll be surprised by how much you can achieve in a relatively short time. I’ll help you to start experimenting on your own and become a successful programmer.

Conventions Used

I use a number of different styles of text and layout in the book to help differentiate between the different kinds of information. For the most part, their meanings will be obvious. Program code will appear like this:

```
int main(void)
{
    printf("\nBeginning C");
    return 0;
}
```

When a code fragment is a modified version of a previous instance, I show the lines that have changed in bold type like this:

```
int main(void)
{
    printf("\nBeginning C by Ivor Horton");
    return 0;
}
```

When code appears in the text, it has a different typestyle that looks like this: double.

I'll use different types of "brackets" in the program code. They aren't interchangeable, and their differences are very important. I'll refer to the symbols `()` as **parentheses**, the symbols `{ }` as **braces**, and the symbols `[]` as **square brackets**.

Important new words in the text are shown in **bold** type.

Code from the Book

All the code from the book and solutions to the exercises are available for download from the Apress web site at <http://www.apress.com>.