

Preface: Why choose this book?

This is a book for people who are interested in learning computer programming, or who are interested in knowing what computer programming is and how computers operate. This book takes the reader quite far into the world of computer programming with the C++ language, but before that, it presents some elementary mathematics related to computers, and introduces the logical operational principles of computers. This book does not assume any previous knowledge about computer programming. So this book is intended for beginners in the field of computing. I have, though, assumed that a person who reads this book has a basic knowledge of mathematics and other subjects equivalent to that provided by an (American) high-school education.

A person studying with this book should have a computer at his or her disposal. The computer should be a personal computer (PC) running a version of the Windows operating system. It is an advantage if the reader is already able to use the computer for purposes like word processing. A connection to the Internet will also be needed on occasion, for when the reader has to download electronic material from the Internet. The reader does not necessarily have to purchase any programming tools for his or her computer. In this book, we use programming tools that are freely available on the Internet.

This book should be easy to read and understand because all computer programs in this book are written with so-called natural names. This means that the computer programs of this book contain a lot of familiar English words and few abbreviations. It has been a tradition in computer programming that programs contain abbreviations like

```
i  j  ch
val char_cnt
str1 str2
```

These kinds of abbreviations can be found in many books of computer programming. In this book, abbreviations like those above have been abandoned. Instead, all computer programs are written with readable natural names like

```
character_index  integer_index  character_from_keyboard
given_value      number_of_characters_read
string_to_search  replacement_string
```

Learning should be easier because the programs in this book are written with readable names like those above. It is true that many other programming books contain programs with long names these days, but I believe that in no other book have the abbreviations been abandoned to an extent such as in this book.

If you have no previous knowledge of computer programming, it may be hard to understand what is actually meant when I say that the programs in this book are written with readable, natural names. You will understand this better when you have learned some computer programming. In the epilogue of this book, I'll tell the full story of how I ended up writing this book. The epilogue clarifies the benefits of this book, but to read the whole epilogue you need to first study the preceding chapters.

The text, figures, tables, and example programs are presented in a unique and most readable manner in this book. The example computer programs are explained graphically with text balloons. Most pages are designed so that you should not turn the pages unnecessarily while you are studying. On some pages you can find out that the lower part of the page is empty of text. The reason for this is that the following page has been made a more readable entity by leaving the preceding page partially empty.

This book should be somewhat entertaining because the example programs in the book have been carefully written and selected. The programs provide useful information from outside the actual field of computer programming. By reading this book, you can find out, for example, where the modern Olympic Games were held in 1956 or who was the president of the United States in 1839.

This book teaches computer programming by presenting the C++ programming language to the reader. Some less important and confusing features of C++ have been left out from the programs of this book. These days, the Java and C# programming languages might be alternative languages to study. I believe, though, that C++ is a fine programming language for a beginner in computer programming, and C++ will be an important programming language for a long time. With C++ you can get "closer to the machine" and learn how a computer operates. With C++ it is easy to write simple computer programs that read something from the keyboard and print something to the screen. These kinds of programs are needed to learn the basics of computer programming. For these reasons, the programming language of this book is C++, but above all, this book teaches computer programming in general terms. Once you have learned C++ and the principles of computer operation by reading this book, you have acquired a robust basis for continuing your studies, and learning other programming languages.

I have written this book because I believe that the way things are presented in this book makes learning easier. I believe also that the way computer programs are written in this book, is a very good way to write computer programs. I have written computer programs this way for more than ten years. Therefore, I think that this book is an excellent way to study computer programming.

Acknowledgments

This book is the result of one man's obsession to write a book that would make the learning of computer programming easier. Although I have worked alone with this book, I am grateful to many people.

First of all, I would like to thank the students with whom I have worked over many years. They have taught me many things related to learning, and because of them I have been able to better write and revise the study materials on which this book is partly based.

Financial help for this work has been provided by the Wihuri Foundation. Veikko Seppänen and Risto Kimari have helped by providing flexible working conditions. Veikko has helped also in many other ways. Ahmad Ibrahim and Neil Rowe have provided me with information regarding the process of book writing and publishing. With Matti Wecström I have been able to discuss all kinds of matters related to this book.

Luke Barker did excellent work while checking the English language in this book. He is not, however, responsible for any errors that are possibly left in this book.

I would like to express my warmest thanks to the people that I have mentioned above. In addition I would like to thank some anonymous reviewers, Juha Alakärppä, Pekka Alaluukas, Cornelia Boldyreff, Attila Feher, Kari Hakkarainen, Olavi Hemilä, Pertti Huuskonen, Veli-Matti Jumisko, Elsa Kallio, Pekka Knuuti, Heikki Kurki, Esa Laamanen, Hannu Laitinen, Veli-Pekka Lehtola, Arto Lukkarila, Terry Lussier, Marko Niskala, Eero Nousiainen, Sauli Ojalehto, Matti Paaso, Markku Peltoniemi, Pekka Rantala, Eija Siivola, Jari Stålnacke, Tuomo Tikkanen, Eija Vieri-Gashi, and Cindy Young. Everyone on the above list has helped me in some way. Over the course of a long writing process, it can be of help if somebody is genuinely interested in your work.

Oulu, Finland
November 2002

Kari Laitinen

How to best study using this book?

Although I have tried my best while writing this book, and I believe this is a very good textbook for a person who is starting to learn computer programming, it is still a fact that computer programming can be a difficult subject to begin with. When you read about something new in this book, you may not understand it immediately. At the start it might be possible that things just won't begin to become clear. That has happened to me, and still I became quite an expert in computer programming. So my first suggestion is

Do not worry if you do not understand something immediately.

It often happens that when you study something that appears to be difficult in the beginning, the difficult thing is only understood later, perhaps the following day when you re-read the text which turned out to be difficult. Therefore, it is of prime importance to learn to accept that there may be difficult things which cannot be understood quickly.

As this book is written so that it should be read sequentially from the beginning towards the end, I advise you to do so. If you encounter something that is already familiar to you, it is up to you if you want to skip some pages. But I would like to remind you, that it may deepen your knowledge, and in any case it does not harm you at all, if you also read the pages that seem to contain something familiar.

While you read have a pencil or pen in your hand and do not hesitate to

- underline or otherwise mark those sentences of the text which you think are important;
- write your own thoughts on the margins and empty spaces of the pages;
- add your own explanation texts near the example programs; and
- mark, for example with a question mark ?, those parts of the text which you did not understand, and which you need to read again later.

Because you learn also by doing, it is important that you write about the things you are studying. Sometimes it may even be helpful to write down the things you do not understand. You can write down on the pages of this book (provided that it is your own book), and you should also have a separate notebook for writing. In addition to writing, it is important that you discuss the matters of computer programming orally with your friends and fellow students. Sometimes it may help you to understand a computer program when you try to explain it to another person.

You just can't learn computer programming without writing programs by yourself. There are plenty of programming exercises in parts II and III of this book. You should do at least part of those exercises with a computer. One possible approach for doing the exercises is that you first read the text of a section and then do the exercises in that section. You should also try to invent your own computer programs. It's nice to work with programs which nobody else has. In Part I of this book there are less exercises, but you should read that part before starting programming in the subsequent parts.

When you try programming exercises, you must use the keyboard of your computer. And in the future, you will probably use a computer and its keyboard daily in your work. To write effectively with a computer keyboard, and to prevent writing mistakes, I recommend that you learn to type with 10 fingers. Computer keyboards are designed so that they can be used effectively with 10 fingers. The skill of typing with 10 fingers can be learned through daily half-hour exercises which last a few weeks. The time you spend to learn that skill will be paid back hundreds of times in the future. There are special computer programs with which you can learn the 10-finger typing system. If you can't find a suitable program, you can borrow an old typing manual from a public library.

While you study this book and do exercises of computer programming, you will notice that you need a lot of information to write computer programs, and you cannot simply remember all the necessary information. The needed information may be in this book

but you do not know on which page the information is located. To find information, you need to learn to use the index that is at the end of this book. The index lists keywords and phrases which are used on certain pages. It is normal that you may not find a certain piece of information by checking the page of your first search word. In these situations you just need to be patient. The index is a useful tool to search information, but you must use ingenuity to invent several search words which can be looked up in the index. This advice applies to the indexes in other books as well.

Some pieces of information are regularly needed when computer programs are being written. For example, you may often have to check things in Appendix A. One possible way to make it easier to find frequently-used data is to take copies of the important pages and hang the copies on the wall near the place where you study or use your computer. Another possibility is to make the frequently-used pages easy to find by attaching pieces of tape or paper clips on them. With these kinds of little arrangements it is possible to make learning and computer programming somewhat easier for you.

The purpose of this book is to be your first book about computer programming. At the beginning this book should be enough. However, as you proceed towards the end of this book, you shall become more and more experienced in computer programming, and you may not consider yourself a beginner any more. You may want to write longer computer programs or try other fields of computer programming. When you get more experienced, I recommend that you also buy other programming books. I, personally, have found out that what you cannot find in one book, can be found in another book. After studying this book, you should be able to read books that are aimed for more experienced programmers. Especially if you start working as a computer programmer, you or your employer should not hesitate to invest in books about computer programming. An appropriate book may easily save many working hours, and the price of the book is usually not much more than the price of a single working hour.

It might be a good idea to re-read this introductory section after you have done some computer programming.