How to Use This Book

In my school days I used to read my science textbooks cover to cover in about a week to ten days from the time they were issued to me. On the other hand, math textbooks took many months to read. Later in life I found when studying a book that was pushing the boundaries of my knowledge that I usually stopped reading after about six or seven chapters and took a few weeks, or even months, off before resuming my study by quickly re-reading the first few chapters and then pushing on with the material that had been overwhelming me first time round and moving on to new material, which would again eventually overwhelm me. I would repeat this process until at the third or fourth shot I would finally finish the whole book.

I suppose that with more self-discipline I would take everything more gently and give myself time to absorb new ideas before pushing on; it just isn't my way. I am always impatient to move on and master new things so I proceed more like the hare than the tortoise.

Which way you learn does not matter as long as you do not suffer from the illusion that acquiring new skills is just a few days' work. Factual textbooks such as those I had for science can be read in a few days or weeks but we are unreasonable to expect to read a book that is designed to help us acquire a new skill in just a couple of weeks.

Another feature of books introducing skills is that they have to assume the reader will practice. It is no good reading a book about playing a flute if you wish to become a flautist. It may be technically possible to read such a book in a few days but that would not turn you into any kind of musician. A single book on flute playing takes many months to read effectively and at every stage you would read the book with your flute readily to hand. You would practice and listen to good flautists.

This book is about acquiring a skill and so I have designed it to be used with a computer to hand. I have also designed it to be studied at whatever pace you feel comfortable with. However I have designed the first seven chapters to work together as a single block. The acceleration from Chapter 1 to Chapter 6 is quite high and most readers will find that they need to take time to digest that material before continuing their studies. I have written Chapter 7 as a natural break before you proceed with the rest of the book.

Each of the next six chapters (8 to 13) is a unit adding some new material and some new ideas. Many readers will find that they want to take breaks after some or even all those chapters. During those breaks you will want to use what you have learnt. Some readers will happily plough straight through in much the same way that they read the first six chapters. That will be fine as long as you set a pace that gives you time to absorb each of the new ideas and practice using them.

During the study of this middle part of the book you should take time out to think about how what you are learning can be used to achieve tasks that interest you. That thinking time is best done away from the book and the computer (much of my thinking is done waiting for buses, enjoying a hot bath or while eating a meal).

Chapters 14 to 16 are different because their main objective is to consolidate your knowledge and skills and show how what you have learnt can be put to use to do things that may look difficult. Roberta comments

that she felt that these chapters treated the reader as if they were now a programmer. She is right, and by that stage in the book you definitely are a programmer, just an inexperienced one. These chapters both show you what can be done with what you know and provide you some useful extras that you can use in your own programming.

The last chapter has the same title as the first exactly because you will have come full circle. You start as someone who has the potential to be a programmer and you finish as someone who knows they can program.

Whether you move from Chapter 1 to Chapter 17 like a hare or a tortoise or in some other way, getting to the end is a new beginning and one where you will truly be able to declare ''I can do it, I can program.''

Study Elements

There are various elements built into this book, all but one of which require your active participation. The exception is that there are places where I give you an anecdote or an analogy to help you with your understanding or motivation.

I use two ways of introducing you to new code. Sometimes I work through developing some code writing about what I am doing at each stage. The purpose is to show you how programs come into existence on a bit by bit basis. During that process I will expect you to work alongside me and create your copy of the program by following in my footsteps.

Sometimes I will give you a finished piece of code and ask you to type it in and correct any typing errors so that the program works. After you have done that I walk you through the code explaining what the pieces do and how they work.

Why the two ways? Sometimes experiencing what a program does greatly aids in understanding how it does it; at other times it is more important to learn how programs come into existence by actually following the thought processes that lead to the finished program. Both ways are valuable to you.

During the course of working through this book you will come across items that are marked as ''tasks''. These are things that you should do before going on with reading. Sometimes they will require you to write a program; sometimes they will simply require that you do something exactly as described. However they share the property that I consider doing them to be an inherent part of successfully reading this book. Sometimes you may eventually have to look at a solution that I have provided but you should think of them as hurdles that you should seriously try to cross without knocking over.

I have also provided exercises. I have tried to choose these so that doing them will help you develop your programming skills without asking you to write dozens of repetitive programs that lead nowhere. In general doing the exercises will be good for you but missing a few will not be a disaster. Your personal pride should motivate you to do the exercises unless they have been marked as ones for specialists (there are a few marked as ''for mathematicians'').

There are some places where I explicitly invite you to try something for fun. These are only a reminder that you should be trying the ideas you find in this book and it should be fun. You should be trying things that you want to show to others. You may be lucky enough to have an appreciative family, friends or colleagues but if you haven't (or even if you have) I want others to see your work and I invite you to send me things you are proud of so they can be made public via the book's website. Your best work will deserve a wider audience so do not be too shy to put it forward.

End of Chapter Elements

Every chapter, bar the last, has an end of chapter section that contains one or more of the following elements:

Roberta's Comments: In which my student author contributes whatever she feels like writing about the chapter in hand. They are an example of something you might consider for yourself: keeping a diary of your experiences. I hope that they will sometimes give you the consolation of discovering that someone else had problems too, and sometimes allow you to feel superior because you didn't. However do not feel too

superior because the text you have is greatly improved over what she learnt from, largely because of the effort she made to criticize my work in a positive way.

Hints: Sometimes I provide a hint for a task or exercise to help you succeed in doing the work yourself.

Solutions: Unlike most books, reading the solutions is not a way of cheating. I expect you to read the solutions when they are provided. Studying the solutions is part of the correct use of the book. Not just reading the solutions or trying them out, but understanding why they work and perhaps why they are different from yours.

Summary: This is broken down under three headings. Key Programming Concepts contains the elements of the chapter that are independent of the programming language. They are the general principles of programming. C++ Checklist gives you a quick summary of the elements of Standard C++ (i.e. the common core of C++ available everywhere) covered in the chapter. And finally there is the Extensions Checklist which summarizes elements that I have added to C++ via the library I provide for you.

End of the Book

I could have added 100 pages to the end of this book by including printed appendices summarizing the C++ language and library, my library, and details of the way the programming style of this book differs from common C++ programming styles. Instead you will find a single printed Appendix A which lists common errors that test readers had when trying to get their code to work.

The other four appendices and the glossary are on the CD that comes with the book. You can print those out if you wish but they will not assist your early efforts when most of their contents will be of no practical use to you.

You may find that there is a greatly extended glossary on the book's website because I plan to add to it in response to questions raised by readers such as yourself. If you meet a term that is puzzling you, check the latest version of the glossary and if it is not there or it still puzzles you, email me and I will do my best to respond promptly and helpfully.

The CD

The CD that comes with this book contains two elements. The first is the software needed by the reader. I will put that more strongly: you should not use programming software that I have not provided either on the CD or on the website; if you do then you are on your own. There are many excellent commercial programming tools available but they are professional tools and as such they are designed to be used by professionals.

The second element is the appendices and glossary. These are provided as Microsoft Word and HTML files. That means that you can print them or use them electronically. The former allows you to add your own annotations and the latter makes it easy to search for a word or phrase.

Installing software from the CD: Unless you have switched off the auto start feature of Windows, the CD should automatically start and lead you through the process of installing all you need. By default it will offer to install in C:\tutorial. If you want to install to another drive just change the drive letter. You can install to a different directory but I would encourage you not to do so. It will make it much easier to follow help provided by others if you have everything in the standard form provided on the CD. You can manage with about 100 megabytes of disk storage but around about 250 megabytes will make it more comfortable and save you from having to clean up intermediate working files from earlier chapters as you progress through the book.

Why fgw?

In many places in this book `fgw` is used as an identifier or prefix. Roberta wondered why, was I dyslexic? This is an example of a little thing that can nag at the back of the mind when we try to do something new. Once we know the reason, however trivial, the irritation goes away. My initials are FWG but I always use fgw to identify my work. The reason is that the school where I taught for almost twenty years identified staff by the initials of their first and last names. Where that left ambiguity the final letter of the surnames was added. There were three members of staff whose initials were FG so I was FGw. I came to feel most comfortable with using fgw as my initials.