

# Book proposal

- Title: **Programming**
- Sub-title(?): A manual for everyone from novice to expert.  
or How to be a programmer.
- Description: Language-neutral practical instruction taking 'never programmed ever' to elite programmer. It combines technical matters (eg this is what SQL is and how you use it) with the mental faculties a programmer needs to develop with a bit of programming culture and dealing with the Real World.
- It sounds impossible but it turns out very well in practice.**
- Style: Coaching one-to-one. A characterful author engages the reader directly in order to build their confidence and commitment to seriously exploring the subject.
- Method: Extremely small steps introducing ideas, then exercises beginning with Javascript. Having established the reader is capable of developing simple programs in their chosen programming language the focus shifts to standard programming matters with the aim of giving the student a good introduction to each aspect. A thread running through the book and developed at the end how to apply intellectual skills.
- Author : Peter Fox  
2 Tees Close, Witham, Essex, CM8 1LG England  
PeterFox=Author@eminent.demon.co.uk  
Tel. (+044) 1376 517206
- Size : 356 A4 pages, 144,000 words
- Format : Text with a handful of simple illustrations.
- Status : **Completed except for a few minor loose ends.**  
[The current draft in PDF format accompanies this proposal](#)
- [Resume](#) page 2
  - [Contents](#) page 3-4
  - [Writing sample](#) pages 1 to 350
  - ['Why you should read this book'](#) page 5...

From the opening page:

*This is a tutorial, starting from the very beginning, on how to be a programmer.*

- *Acquire the skills of writing computer programs*
- *Find out what makes programmers different*
- *Learn how to avoid the mistakes ordinary programmers make*
- *Develop your of creativity and concentration*
- *See the world afresh in the crystal sharp focus of precision analysis*

*Whether you're thinking of trying some simple programming, already do a bit, get paid to program or work with programmers this book will open your eyes and put you on the right track.*

*This is not a tome on Computer Science, and doesn't set out to teach a particular language. The ultimate objective is to show how to become an elite programmer, someone who can 'see through walls', asks the right questions of the right people, produces sleek and robust programs...*

*... and be well paid, well respected and enjoying a superior lifestyle.*

**Different, comprehensive, complimentary and capturing the essence.**

This book is unique in that it covers the *whole subject as a tutorial*. (Amazing! I thought it was impossible.) It hasn't been done before and I believe it is comprehensive enough to remain definitive.

There are lots of *specific method* 'typewriting for authors' books, from 'lean foo in a day' to 'technical techniques in detail', but where is the *what is it all for* and *how does it all hang together* and *what are the core skills* that applies to programming in general book?

- Not only does this book fill the need for a unified approach to the description of the art of programming,
- but it also fills the need for a practical, language independent, beginner's tutorial ...
- ... with the advantage of continuing through to what makes an elite programmer.

There are general computer science and course-related texts but this is rather different. In fact this is meant to be complimentary to CS and specific language texts.

- The reader is given the practical background and key concepts necessary to investigate, evaluate and use computer science.
- Many readers will be learning one or more programming languages in parallel. This book shows them how to apply their language knowledge effectively.

The title could be 'how to be a programmer' as that's what the book covers. There is plenty of the usual technical stuff but a theme develops along the lines of 'why programmers are different'. This is the magic extra ingredient that turns 'become a programmer in 20 easy steps' into 'get the buzz that comes from being a hot shot'.

I've managed to create a single, unified career escalator in a style addressed directly to the reader, finding a balance between detail/depth and pace/progress. The practical technicalities are mingled with the 'essence of being a programmer' and a bit of programming culture which gives the reader confidence that they will soon be an 'insider'. That's quite a tricky mix to achieve.

### **Audience and why they'd purchase**

The immediate audience is people who want to know generally what programming is about from a practical point of view.

- Absolute beginners
- Potential and recent recruits to the profession - including students
- Curious dabblers looking to find out what the rest of the iceberg is like

There is a separate audience of current professionals who probably don't believe such a book can be written, who are generally interested in the subject (the same way steam engine buffs buy steam engine books), the hooks here are:

- What makes an elite programmer?
- Is this book a good buy for their nephew?
- Should there be elements extracted for the benefit of lesser colleagues and managers?
- The importance of programmers as highly skilled professionals not simply code-monkeys.

*So that means just about anybody interested in programming and employing programmers!*

The reasons for purchase are:

- To learn : Practical skills of programming
  - Filling an ambition to quickly get to join a well paid elite
  - One book gives the full map for developing skills
  - It's a practical start on a profitable career
- To develop : What qualities make top class programmers
  - It's a unified explanation of why programmers are different ( - and better!)
- For entertainment : Chatty exposure of technical cock-ups, organisational and personal failings
- For interest : Ah yes - That sums up what I sort-of-knew in a nutshell.

Bonus book characteristics:

- It puts a technical skill (eg a specific programming language) into a real-world context.
- It is not an impersonal review of 'programming' but a one-to-one 'how you too can be one of these super-clever people'.

The book buyers

- Anyone who buys a language specific book should be interested in this.  
*(It works the other way round as well : Having discovered through this book that programming is straightforward providing you make a little time, then readers will go and buy a language specific book.)*
- Many mature professionals would be interested as a book to give or recommend to family and other people who are thinking about IT as a career.
- This book is not really suitable for use as a college course on programming because it is focussed on *individual* progress, understanding and enlightenment. However it could be valuable additional or preparatory reading for programming courses and for situations where programming is ancillary to the main activity. It should certainly be found in quantity in college libraries.

### **Marketing notes**

There is no particular window of opportunity.

The niche being aimed for is 'classic' (admittedly a tough call) and doesn't fit into any series or category. There is a history of computing classics being only recognised by aficionados which this book can overcome by being immediately accessible to complete beginners.

There are some 'hooks of chattering interest' such as

- *What Could Possibly Go Wrong? Just as clocks go 'tick' and cows go 'moo' so programmers go 'wcpgw?'*. "WCPGW" is used as a buzz-word throughout.
- Beginner : A semi-formal human instruction language
- Fudge : A 'language' that looks enough like any mainstream programming language for easy transcription into the reader's chosen language.
- The final chapter: "Get a life!" which puts the intellectual and self-discipline threads into everyday work situations.

Bill material

- This is the book absolute beginners need to discover
  - (a) a bit about actual programming (the easy way) to start with, and then
  - (b) the fascination of programming and why programmers are 'different'
- This is the book 'I picked up a bit of programming' people need because
  - (a) it lays into bad habits and shows how to do the job professionally.
  - (b) it gives a guide to how programming fits into the Real World
- This is the book that programmers should read because
  - (a) it puts a lot of vaguely understood matters into clear focus
  - (b) it shows the difference between a code-monkey and a top-flight programmer
- This is the book non-programmers and people who work with programmers should read because
  - (a) everyone can benefit from the clear, logical, sceptical, mixture of what-if speculation and precision mental processes used by programmers.
  - (b) it will give an insight into why programmers are 'different' and how they can be managed, developed and used to advantage.

### **Production notes**

The text has been developed without figures for convenience. There are opportunities for a few illustrative diagrams.

The manuscript has been produced using a variety of colours. The distinctions are necessary but with professional typography and design could be rendered in black and white.

There are a few loose ends remaining to be tidied and checked. There is no index or proper contents.

There is no intention of providing an accompanying CD or web content.