

Free Book Review: How to Think Like a Computer Scientist

Reviewed by Hong Feng
fred@mail.rons.net.cn

Jan 01,2002

When I first time met Java in 1995, there was only a few books to read about it, David Flanagan's *"Java in a Nutshell"* is perhaps the best one I read and translated into Chinese, it well documented the Java 1.0, including the core language plus more than 200 built-in classes came with JDK 1.0, the book is good for C/C++ programmers, as it gave a detailed explanation on how C/C++ programmers immigrate to Java.

Java was designed as a system programming language, in many aspects, it is a modern derivation of C. Though Microsoft is working hard on its .Net project with C#, but Java already took a huge step ahead of it. Java's specification is open, and there are already a lot of support to it from our free software community — from JVM (Java Virtual Machine) like Kaffe, to compiler like GCC 3.0, on GNU/Linux system, now you could write free software without using any proprietary tools from SUN Microsystems (— well, I shall admit I am a little bit exaggerated here, but for most cases, that's the case.)

However, David's book (and his Java books later came out on Java) are proprietary, that means you could not legally copy and redistribute it freely, his publisher may sue you violate the copyright license. Also his book is written for C/C++ programmers only, not for beginners who want to learn Java from scratch.

Fortunately, today there are many nice books which are free for you. Prof. Allen Downey's works entitled **"How to Think Like a Computer Scientist"** is fairly good one of them, and it is tru-

ly free under **GNU FDL (Free Documentation License)**, which is freely downloadable from <http://www.welleslet.edu/downey/ost>.

Actually, Prof. Downey has written several books for *"How to Think Like a Computer Scientist"* as a series, and Java version is one of them (the rest two I have read are about Python and C++, which are also free under GNU FDL, but this review is just focused on the Java version).

Most books on programming will be quite overwhelming for beginners, so the readers usually get headaches to learn it, this is specially the true case for Java programmers, as today, Java is huge in size, SUN Microsystems documented it in more than thousands pages in their official documentations. It is even very hard to summarize the main features of it.

Allen's book does not go this way, instead, he's set his goal of the book as to be used as truly textbook for computer science department students, and tried to demonstrate the aesthetic part of Java programming. In the book, he tried to lead his students to treat computer scientist as a combination of roles of "mathematician, engineer and natural scientist", which means the reader is expected to act like the mathematicians to donate ideas (specifically those of computation problems), like the engineers to design things and assembling components into a system and evaluation tradeoffs among alternative (e.g. performance, price, time from design to delivery ...), like the natural scientists to observe the behaviors of a complicated system, form hypotheses, and test predictions.

The start level of the book is almost zero, so anyone who wants to learn Java programming could use this book for their studies. The contents goes well in heuristic way from the beginning to the end.

Usually, the author starts from a simple contents based on the knowledge introduced in the previous chapters, then gets more new stuffs in. For example, Java is an OOP (Object Oriented Programming) language, the book introduced variables and types in Chapter 2, and methods in Chapter 3, object in conception in Chapter 8, and finally the OOP in Chapter 13.

Also, as the book is aimed to use by computer science students, so it naturally covered the topics of data structure, and the Java's way to operate on the data structures including arrays, lists, stacks, queues, trees, heaps and hashes. It gives out the algorithms in Java code to operate on them, means when you really know how they work out, you could start the programming in your real world project.

By the end of each chapter, there is a glossary to go over the terms newly introduced in this chapter, which is very good and helpful for students to review what they have learnt in this chapter.

Like any other books, the book has some places to improve, for example, the Chapter 4 covered the graphics, which is a very interesting topic in Java programming, but the book has not mentioned the Java's 2D features in short. Also the book should give the students a roadmap on how to read the SUN's documentation on Java, so that students could go there are start to program as soon as they learn this book. So I suggest here that Prof. Downey to include them when he upgrade it into the next edition.

The Table of Content is missed from the downloadable L^AT_EX file over the web site. Also it would be much easier for beginners to have a single Postscript file or a PDF file to download from the site, as some web visitors might not know how to work out from the L^AT_EX files from the scripts built-in.

Despite of these, the book was proofread by many professionals, as it is heavily concentrated on how to increase your ability to solve the real problems with the necessary theory clearly explained, so it is a great choice for student who wants to become a truly professional Java programmer who could think like a truly computer scientist, which is a basic requirement and important when one hacking some free software projects with Java.

Reference Resource:

- Title: How to Think Like a Computer Scientist
- Author: Allen B. Downey
- Publisher: Green Tea Publishing, Inc.
- URL <http://www.welleslet.edu/downey/ost>
- Copyright Status: Free, under GNU FDL

About the Reviewer: Hong Feng is the publisher of **FREE SOFTWARE** magazine which you are reading, he started hacking on GNU/Linux in 1995 with 30 floppy discs. As a programmer, he develops device drivers for hardware, Sometimes he also joins his typesetting workshop in Wuhan to produce high quality papers and books with T_EX.

He launched **M_NM Project** on March 05, 2001 for supporting the free software community in China. He often visits universities and gives speeches about free software to the students, discusses with professors on curriculum reform for computer science departments, and find programmers for M_NM Project. He offers training courses for free software tools. Sometimes he acts as a “doctor in clinic” for free software companies and developers to help them to grow their business by giving practical consulting advice.

In his spare time, he likes reading (especially books about history), studying philosophy and mathematics theory. He can be reached by email fred@mail.rons.net.cn or hongfeng@gnu.org