"A MUST READ FOR EVERY PROGRAMMER" PETER NORTON

THE PROGRAMMER'S JOB HANDBOOK

THE SKILLS YOU NEED FOR LONG-TERM JOB SECURITY AND PROGRAMMING SUCCESS

GENE WANG

Executive Vice President of Applications and Development Tools at Symantec Corporation

AN INDISPENSABLE GUIDE FROM AN INDUSTRY EXPERT

The Programmer's Job Handbook

Gene Wang

Osborne/McGraw-Hill

Berkeley New York St. Louis San Francisco
Auckland Bogotá Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

Osborne **McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

The Programmer's Job Handbook

1234567890 DOC 99876

ISBN 0-07-882137-1

PUBLISHER *Lawrence Levitsky*
AQUISITIONS EDITOR *Scott Rogers*
PROJECT EDITOR *Emily Rader*
COPY EDITOR *Judy Ziajka*
PROOFREADER *Linda Medoff*
COMPUTER DESIGNER *Peter F. Hancik*
ILLUSTRATOR *Randall Marsden Design*
QUALITY CONTROL SPECIALIST *Joe Scuderi*

Acknowledgments

During the time that I spent writing this book, I was continually gratified by the support I received from friends in the industry, all of who wanted to contribute something to help programmers succeed in their careers. I sincerely wish to thank the many friends and colleagues who helped by contributing their wisdom. At Symantec, I talked to dozens of people, including Gordon Eubanks, Walter Bright, Bill Coleman, Mansour Safai, Steve Singh, Enrique Salem, Gary Barbato, Peter Norton, and Pat Vermont. Brad Silverberg and Jim McCarthy of Microsoft were tremendously supportive and helpful, as was Jerry Barber of Adobe, Dan Putterman of Maximum Information, Jeffrey Tarter of *Softletter 100*, Paul Gillin of *Computerworld*, Bruce Brereton of Novell, Anders Hejlsberg of Borland International, Githesh Ramamurthy of CCS, and Yulun Wang of Computer Motion. Bill Donovan and Susan Yun added a lot of insight from their recruitment experience successfully placing programmers in high-paying jobs. My mother, Jo Szeto, helped out by continually reminding me that she was an English major. And I'd like to thank Scott Adams, the creator of *Dilbert*, for letting me use some of his *Dilbert* comics in this book, for sharing his insights on choosing the right company, and for enriching my life every day with his humor. I especially wish to thank Janel Garvin for helping out with this project, and my editor at Osborne/McGraw-Hill, Scott Rogers. Thanks also to Emily Rader and Daniela Dell'Orco.

Introduction

Programming is a great career. Technology is changing the shape of our civilization from one year to the next, creating great new job opportunities with high financial rewards and a multitude of software specialty areas to master. Computer software and the programmers who create it are literally at the heart of the information age and all the new opportunities that it brings. Truly, programming is one of the most exciting professions to pursue as we head into the twenty-first century. But you have to carefully plan and manage your career path to aim it on a high-growth trajectory. If you master the fine art of programming, you will have a highly desired skill set and a bright job future to look forward to. This book is meant to help programmers, technical managers, software analysts, quality assurance engineers, and all others associated with software development, to find that right path to the future.

Together, we will explore the programmer's job market of tomorrow by examining in depth the job market as it exists today. We'll take a look at what programming skills are in demand and how much different companies are paying by industry and geographical region. I boldly predict seven software technologies that I think will be hot in the future and hope that I'm right **;-)**. In addition, from interviews with many of the top software industry experts, I provide a list of seven skills of successful programmers. Hopefully, by reading

this book, you will be able to optimize your personal career path in life and improve your job selection process to find and get the best jobs.

This book is not a technical book. It is not designed to show you any specific tricks and techniques for writing good code or designing software architectures. There are plenty of excellent books around on those topics, and there are also classes and conference sessions available to help you get the technical expertise you need to succeed. This book is designed to develop your skills in managing your career path in order to put you in line for the jobs that will be the most rewarding and fulfilling as you progress through your career.

Are you aware that programmers in the United States on average make more than twice as much as the average American? Did you know that programmers with a knowledge of C++ command better salaries and get jobs a lot easier than programmers with other language specialties? Did you know that Microsoft Foundation Classes (MFC) is the framework that you need experience with in order to get most of the programming jobs for creating desktop software? Did you know that surfing the Net is one of the fastest, easiest, and cheapest ways to find the perfect job?

How This Book Is Organized

In Chapter 1, I talk about what makes a successful programmer. Next, in Chapter 2, on the programmer's job market, I arm you with the statistical information you need to shape your career in the direction you want it to go.

Once you're oriented to the programming job market overall, Chapter 3 provides some nitty-gritty, down-to-earth advice on actually getting the job you want. Unfortunately, it's not enough to be a great programmer. To get a great job you have to also know how to write a résumé, how to conduct yourself in an interview, and where to look to find the positions that are available. This chapter gives you plenty of inside tips on getting the job you really want. It talks about how to prepare for an interview, how to create a dynamite résumé, when to use recruiters, and even how to know when it's time to leave your old job and move on. It also includes a checklist of items to review before you go into an interview, to help you when you're at a new employer's office trying to convince them you're right for the job. This chapter provides you with tools to assess yourself and your career in order to determine how you're doing at providing yourself with the career you want and deserve.

Chapter 4 outlines seven skills of successful programmers and introduces the idea of career management, as opposed to simply getting a job. It discusses

how to manage your career both when looking for a different job and when executing the job you have now. The advice here comes from years of experience—my experience and the experience of software development vice presidents and directors from such companies as Microsoft, Apple, Adobe, and Symantec.

Chapter 5 covers the subject of choosing your programming tools. Great programmers need the best tools. I have dedicated much of my professional career to building great development tools, starting in 1983 with an AI programming tool called Golden Common LISP, then building Borland's C++, and now Symantec's award-winning Symantec C++. This chapter lists some of the tools available and I give you some pretty good insider advice on selecting tools that are powerful, versatile, and that save you time and effort.

Chapter 6 explores seven hot technologies for programming careers. From the Internet and networking, to Windows 95 and multimedia, to client-server development and distributed computing, some of the fastest growing technology trends lead to the biggest job opportunities.

Chapter 7 discusses component programming, which to me is one of the most exciting technologies since the introduction many years ago of object-oriented programming. I believe this is one of the most important technologies for programmers to know to get ahead in the future. Component programming will form the cornerstone of rapid application development in the next 10 to 15 years.

In Chapter 8, I talk about optimizing the software lifecycle. The challenge of delivering high-quality software on time has never been higher. Here I discuss a process for software project management used successfully at Symantec.

Finally, in Chapter 9, I highlight some key factors in picking the right organization to work for and list the top 100 software companies today.

Writing this book has been a real learning experience for me. I hope that reading the assembled information will be rewarding for you and put you on the right path to a brilliant career.

Contents

1 The Successful Programmer

High technology is one of today's fastest growing fields. The need for software is accelerating as the information age moves into the twenty-first century. In virtually every industry, in every country around the world, organizations need programmers. This creates great job opportunities for programmers who are successful. The natural question is how do you become a successful programmer? As you might expect, there is more than one answer.

Albert Einstein said, "Try not to become a man of success, but rather, try to become a man of value." Being good at programming is a prerequisite to job success. One way to amplify success is to gain unique expertise in technical areas that many companies are focusing on. For example, if you become an expert in Windows 95 programming, client-server applications, usability testing, embedded systems development, or object-oriented programming, your career opportunities will improve considerably. This book discusses hot technology areas for programmers in more depth in Chapter 6, "Hot Technologies for Programming Careers," and Chapter 7, "Component Programming." Suffice it to say here that developing your skills in the following areas will increase your value as a programmer:

- Windows 95 programming
- Object-oriented programming

- Multimedia
- Internet
- Networking
- Wireless communications and mobile computing
- Client-server development

Measures of Success

There are many different ways to measure success. Money is a common metric. The highly successful programmer gets paid more than his or her peers. The average programmer who has been in the profession for a few years makes just over $54,000 a year. Senior programmers can make more than $80,000, and many lead programmers or architects writing significant programs can earn well into the six-figure range. A survey by *Fortune* magazine (June 26, 1995), summarized in Table 1-1, compares the entry level and average salary of programmers with the salaries of several other professional occupations.

In addition to salary, a person often receives bonuses and other recognition for work well done. Benefits such as retirement plans, stock options, extensive health-care coverage, and generous vacation time are also common in the programming field.

As you will see in Chapter 2, "The Programmer's Job Market," salary levels vary a great deal depending on where you work and what industry you choose. Being in the right place at the right time is a key ingredient of success. Although there are programming jobs in virtually every industry, some industries pay better than others. For example, on average you'll be better rewarded financially in banking or real estate than in government or education. Also, of course, while

	Entry Level	Average Salary
Software engineer	$33,702	$54,470
Accountant	$24,750	$36,500
University professor	$39,050	$49,490
Government economist	$21,486	$46,852
IRS tax auditor	$19,500	$38,500

Table 1-1 *Average Salaries for Programmers and Selected Other Professionals*