

"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.

For information on translations or book distributors outside the U.S.A., or to arrange bulk purchase discounts for sales promotions, premiums, or fundraisers, please contact Osborne **McGraw-Hill** at the above address.

The Programmer's Job Handbook

Copyright © 1996 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

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*

Information has been obtained by Osborne **McGraw-Hill** from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Osborne **McGraw-Hill**, or others, Osborne **McGraw-Hill** does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from use of such information.

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	<i>The Successful Programmer</i>	1
	Measures of Success	2
	The Path to Successful Results	4
	The Software Lifecycle: A Blueprint for Successful Results	5
	Successful Programmers	7
	Managing Your Career	8
	Ten-Year Career Plan	10
	When to Make a Change	10
	Attaining Success	11
2	<i>The Job Market for Software Developers</i>	13
	The Overall Job Market	14
	Industry Segments	15
	International Opportunities	16
	Types of Programming Jobs	19
	Corporate Programming	19

Shrink-Wrapped Software Development	21
Pure Research and Development	22
Contracting	23
Embedded Systems Programming	25
Client-Server Programming	26
Database Programming	27
Other Fields	27
Jobs by Language and Platform	28
Industries for Programmers	28
Platforms Today and Tomorrow	29
Speaking the Language of Jobs	35
Job Titles and Categories	37
Internationalization Engineer	39
Quality Assurance Engineer	39
Software Engineer—UNIX	39
Software Engineer—Database	39
Software Engineer	40
Software Engineer—Internet	40
Technical Support Engineer—Networks	40
Technical Writer	40
Software Quality Assurance Engineer	41
Release Lab Engineer	41
Quality Assurance Engineer—Japanese	41
Software Engineer—Localization	41
Technical Support Analyst—International, Central Europe	41
Quality Assurance Supervisor	42
Development Manager	42
Software Engineer—UI Specialist	43
Quality Assurance NLM Engineer	43
Development Manager—Network Communications	43
Senior Quality Assurance Analyst	44
Senior Development Director	44

Software QA Engineer	45
Associate QA Engineer I	45
Software Quality Assurance Manager	45
Principal Software Engineer	46
Senior Technical Writer	46
Senior Manager, Product Marketing	46
Senior SQA Analyst	47
Software Architect	47
Computer Operator	47
Software Tools Engineer	48
Technical Writer Specialist	48
Where the Money Is	48
Forms of Compensation	51
Women in Programming	57
Future Outlook	57

3	<i>Getting the Job: The Perfect Résumé and Interview</i>	61
	How Jobs Become Available	61
	How Companies Recruit Applicants	63
	Look Online for Jobs	65
	The Perfect Résumé	66
	Creating a Résumé That Positions You in the Best Way	67
	Major Internet Sites for Jobs	67
	Matching Your Résumé to the Job Description	71
	Analyzing the Sections of Your Résumé	75
	The Perfect Interview	76
	Preparing for Phone Screening	77
	Preparing for the Interview	77
	The Interview	78
	Second-Round Interviews	81
	References	82

Working with Recruiters	82
How to Find a Recruiter If One Doesn't Find You	83
Using Employment Agencies	84
A Preinterview Checklist	85
Before You Leave for the Interview:	85
Review Your Responses to These Commonly Asked Interview Questions:	86
Tips for the Interview:	86
Questions to Ask the Interviewer:	86
How and When to Leave Your Current Job	87
Leaving a Company	89
The Law in Hiring Practices	90
Following Up	92

4	<i>Seven Skills of Successful Programmers</i>	95
1. Underpromise and Overdeliver		95
2. Don't Put Bugs in Your Software		97
3. Be Passionate and Hard Working		100
4. Know What You Don't Know		101
5. Work Well with Your Team		104
6. Follow Through to the End		106
7. Keep Up with Emerging Technology		108

5	<i>Choosing Your Tools</i>	111
Languages		112
Integrated Development and Debugging Environments		114
Project Management System		115
Other Tools to Make Your Build Process More Efficient		115
Object-Oriented Editors and Browsers		116
Debuggers		117

Visual Tools	118
Stand-Alone Editors	119
Class Libraries	119
Fourth-Generation Language Tools	121
Add-Ons to 4GL Tools	122
Version Control Tools	123
Testing Tools	123
Choose Your Tools Carefully	124

6 *Hot Technologies for Programming Careers* 125

Windows 95 Programming	128
Object-Oriented Programming	129
Multimedia Programming	130
Digital Cameras and Graphics	132
Programming for the Internet	133
Data Security	134
Electronic Commerce	135
Networking	137
Mobile Computing and Wireless Communications	138
Client-Server and Distributed Computing	140

7 *Component Programming* 143

Object-Oriented Programming	144
Microsoft Foundation Classes (MFC)	148
Object Linking and Embedding (OLE)	149
OpenDoc	152
OpenDoc Services	152
Common Object Request Broker Architecture (CORBA)	153
System Object Model (SOM)	154
The Future of Component Programming	155
The Pasta Theory of Programming	155

8	<i>Optimizing the Software Lifecycle</i>	161
	Concept Phase	163
	Tools for Understanding Your User	164
	Proposal Phase	167
	Common Questions for a Development Project	168
	Definition Phase	169
	Build Phase	172
	Test Phase	176
	Master Phase	178
	Ship Phase	179
	Maintain Phase	180
	Optimizing Development with a Software Lifecycle	180

9	<i>Selecting the Right Company for You</i>	181
	Seven Tips to a Successful Career	182
	Specialize in a Platform with a Future	182
	Master the Right Tools	183
	Identify a Solid Development Strategy	183
	Pick the Right Product	184
	Have a Career Plan	185
	Choose an Organization You Like	185
	Establish Balance in Your Life	187
	The Top 100 Software Companies	187

A	<i>Internet Job Search Fundamentals</i>	223
	Getting Connected	224
	Connecting via a Dedicated T1 or T3 Line	224
	Connecting via a Commercial Service	225
	Connecting via a PPP or SLIP Line	227
	Exploring the Internet	227

Internet Addresses	227
Company Sites	229
Job Sites	229
Newsgroups	229
Gophers	232
Yahoo	233
Responding to an Internet Job Listing	236
Getting the Job Electronically	237
Using the Internet to Find Jobs in Computer	
Hardware	237
Adaptec	237
Advanced Logic Research	238
Advanced Micro Devices	238
APC	238
Apple Computer	238
BusLogic	238
Compaq	239
Data Exchange	239
Data General	239
Dell	239
Design Acceleration	239
Digital Equipment Corporation	240
Digital Products	240
Direct Data Storage	240
Encore Computer	240
Gilltro-Electronics	240
Heurikon	241
Hewlett Packard	241
IBM	241
IBM Personal Computers	241
Intel	241
National Instruments	242
NEC	242

NECX Direct	242
Network Wizards	242
Online Computer Market	242
Overland Data	243
PC Market	243
Play, Inc.	243
Power Computing	243
RICOH California Research Center	243
Road Warrior International	244
Seagate	244
Silicon Graphics	244
Sparco Communications	244
Standard Microsystems	244
Sun Microsystems	245
Supra	245
Texas Instruments	245
Toshiba America	245
TouchWindow	245
U.S. Robotics	246
WebFactory	246
Well-Connected Mac	246
Wiltec	246
Xerox	246

B *More Great Sources of Information for
Your Programming Career* **247**



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

Fortune Magazine, © 1995 Time Inc. All Rights Reserved.

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