

***Beginning
Programming***
ALL-IN-ONE DESK REFERENCE
FOR
DUMMIES®

by Wallace Wang



WILEY

Wiley Publishing, Inc.

***Beginning
Programming***
ALL-IN-ONE DESK REFERENCE
FOR
DUMMIES®

by Wallace Wang



WILEY

Wiley Publishing, Inc.

Beginning Programming All-In-One Desk Reference For Dummies®

Published by

Wiley Publishing, Inc.

111 River Street

Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, the Wiley Publishing logo, For Dummies, the Dummies Man logo, A Reference for the Rest of Us!, The Dummies Way, Dummies Daily, The Fun and Easy Way, Dummies.com, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 800-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002.

For technical support, please visit www.wiley.com/techsupport.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2008927908

ISBN: 978-0-470-10854-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1



WILEY

About the Author

I started off as a writer and wound up becoming a computer programmer. Then I wound up circling around again to become a writer about computers. I've spent most of my life writing about and programming a variety of personal computers ranging from an ancient PC running MS-DOS 1.25 to Windows XP PC to the latest Macintosh computer running Mac OS X Leopard. My only preference for any computer is to use one that works.

I first learned about programming from my high school's ancient teletype terminal that connected to a mainframe computer through a 300 baud acoustic modem that often disconnected me in the middle of my BASIC programming sessions. At the time, I didn't know much about programming. I just taught myself BASIC from a book and illegally gained access to the teletype terminal by using somebody else's password. Later in the year, I actually signed up for a computer class and finally gained legitimate access to the teletype terminal to do everything I had been doing illegally long before.

The first time I wrote a BASIC program on my own, it was a game that simulated flying a nuclear-armed bomber through a variety of anti-aircraft defenses including surface-to-air missiles and jet fighters trying to shoot you down. When this program worked for the first time, I felt like Dr. Frankenstein watching his creation twitch and come to life. To this day, I still experience that same feeling of exhilaration in creating something from an idea and turning it into an actual working program. Only other programmers can understand this strange sense of power and elation that comes from a working program, and it's this same sense of wonder and exploration that I hope you'll experience as you use this book to explore the world of programming on your own computer.

I may be considered a computer veteran after all these years, but that doesn't mean that I can't still experience that same feeling of satisfaction in typing that final command and watching an entire program work exactly as I wanted. Although I've written plenty of other books both on computers (*Microsoft Office 2007 For Dummies*) and far away from computers altogether (*Breaking Into Acting For Dummies*), I find that programming still fascinates me to this day.

As an author, I hope to help you discover your own path to learning programming, and as a programmer, I hope to provide an overview of computer programming in general. You may not become an expert programmer after reading this book, but if you come away with a greater appreciation for programming, then I'll know I'll have fulfilled my duty as both an author and programmer for this.

Dedication

This book is dedicated to anyone who wants to learn how to program a computer. Computer programming can be one of the most creative ways to express your ideas so if you have your heart set on writing programs for fun or profit, you've just joined a select group of fellow renegades, entrepreneurs, and hobbyists who find programming an enjoyable intellectual exercise. When lost in the world of programming, you can often elevate your spirit to lofty heights of pleasure and wind up crashing right back down to Earth again when a single syntax error causes your program to crash an entire computer. Welcome to the wonderful world of programming. You deserve to achieve whatever your mind can envision and your programming skills can create.

Acknowledgments

This is the part of the book that most people skip over since it usually lists a bunch of names that most people have never heard before, so before you can skip over this page, I'd like to thank you for buying (or at least reading) this book. If you're interested in learning to program a computer, you've already separated yourself from the masses who are ecstatic when they can just get their computer to work in the first place. As a programmer, you have the power to control how people may use computers in the future, and this power can give you the chance to help others or make someone completely helpless in their agony when trying to use a computer, so use your programming skills wisely.

On another note, this book owes part of its existence to Bill Gladstone and Margot Hutchison at Waterside Productions and another part of its existence to Katie Feltman at Wiley Publishing for turning this project into reality. Some other people who helped shape this project include Pat O'Brien and Vince McCune.

I also want to acknowledge all the stand-up comedians I've worked with over the years including Darrell Joyce (<http://darrelljoyce.com>), Leo "the Man, the Myth, the Legend" Fontaine, Chris Clobber, Bob Zany (www.bobzany.com), Russ Rivas (<http://russrivas.com>), Don Learned, Dante, and Dobie "The Uranus King" Maxwell. Another round of thanks goes to Steve Schirripa (who appeared in HBO's hit show "The Sopranos") for giving me my break in performing at the Riviera Hotel and Casino in Las Vegas, one of the few old-time casinos that hasn't been blown up to make way for yet another luxury hotel and casino targeting rich people who want to lose their money faster.

Additional acknowledgements also go to my fellow radio co-hosts: Rick Gene, Drizz, Justin Davis, and Dane Henderson, who shared the airwaves with me during our year at 103.7 FreeFM in San Diego. Having our own radio show gave us all an excuse to go places and do things that we wouldn't normally get to do such as visiting a UFO cult that built a landing strip for a flying saucer, exploring a museum that promotes creationism, and visiting Comic-Con where people dress up as their favorite comic book super heroes. (I tried to give ten bucks to anyone who would let me pour gasoline on them and light a match so they could attend Comic-Con dressed up as the Human Torch, but nobody was willing to take me up on that offer.)

I'd also like to acknowledge Cassandra (my wife) and Jordan (my son) for putting up with my long hours and my insistence that everyone dump their Windows XP PCs right away (an easy decision since they never worked right anyway) and migrate completely to the Macintosh. Final thanks go to Bo, Scraps, and Nuit (my cats) along with a final farewell to Tasha, who was a special breed called a Korat. R.I.P.

Publisher's Acknowledgments

We're proud of this book; please send us your comments through our online registration form located at www.dummies.com/register/.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Project Editor: Pat O'Brien

Acquisition Editor: Katie Feltman

Copy Editor: Jen Riggs

Technical Editor: Vince McCune

Editorial Manager: Kevin Kirschner

Media Project Supervisor:
Laura Moss-Hollister

Media Development Specialist: Angela Denny

Editorial Assistant: Amanda Foxworth

Sr. Editorial Assistant: Cherie Case

Cartoons: Rich Tennant
(www.the5thwave.com)

Composition Services

Project Coordinator: Patrick Redmond

Layout and Graphics: Claudia Bell,
Reuben W. Davis, Melissa K. Jester

Proofreaders: Laura Albert, John Greenough,
Jessica Kramer, Christine Sabooni

Indexer: Ty Koontz

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Acquisitions Director

Mary C. Corder, Editorial Director

Publishing for Consumer Dummies

Diane Graves Steele, Vice President and Publisher

Joyce Pepple, Acquisitions Director

Composition Services

Gerry Fahey, Vice President of Production Services

Debbie Stailey, Director of Composition Services

Contents at a Glance

<i>Introduction</i>	<i>1</i>
<i>Book I: Getting Started</i>	<i>5</i>
Chapter 1: Getting Started Programming a Computer	7
Chapter 2: Different Methods for Writing Programs	29
Chapter 3: Types of Programming Languages	49
Chapter 4: Programming Tools	83
Chapter 5: Managing Large Projects with Software Engineering.....	107
<i>Book II: Programming Basics.....</i>	<i>125</i>
Chapter 1: How Programs Work	127
Chapter 2: Variables, Data Types, and Constants	141
Chapter 3: Manipulating Data	161
Chapter 4: Making Decisions by Branching	181
Chapter 5: Repeating Commands by Looping	199
Chapter 6: Breaking a Large Program into Subprograms	213
Chapter 7: Breaking a Large Program into Objects.....	231
Chapter 8: Reading and Saving Files	259
Chapter 9: Documenting Your Program.....	277
Chapter 10: Principles of User Interface Design.....	289
<i>Book III: Data Structures</i>	<i>309</i>
Chapter 1: Structures and Arrays.....	311
Chapter 2: Sets and Linked Lists	329
Chapter 3: Collections and Dictionaries.....	345
Chapter 4: Stacks, Queues, and Deques	361
Chapter 5: Graphs and Trees	375
<i>Book IV: Algorithms.....</i>	<i>391</i>
Chapter 1: Sorting Algorithms	393
Chapter 2: Searching Algorithms.....	409
Chapter 3: String Searching.....	423
Chapter 4: Data Compression Algorithms.....	435
Chapter 5: Encryption Algorithms	445

<i>Book V: Web Programming</i>	461
Chapter 1: HyperText Markup Language	463
Chapter 2: CSS	477
Chapter 3: JavaScript	485
Chapter 4: PHP	497
Chapter 5: Ruby	509
<i>Book VI: Programming Language Syntax</i>	521
Chapter 1: C and C++	523
Chapter 2: Java and C#	541
Chapter 3: Perl and Python	559
Chapter 4: Pascal and Delphi	575
Chapter 5: Visual Basic and REALbasic	589
<i>Book VII: Applications</i>	607
Chapter 1: Database Management	609
Chapter 2: Bioinformatics	625
Chapter 3: Computer Security	633
Chapter 4: Artificial Intelligence	643
Chapter 5: The Future of Computer Programming	657
<i>Index</i>	671

Table of Contents

Introduction..... 1

Who Should Buy This Book	1
How This Book is Organized.....	2
Book I: Getting Started.....	2
Book II: Programming Basics	2
Book III: Data Structures	2
Book IV: Algorithms	2
Book V: Web Programming.....	3
Book VI: Programming Language Syntax.....	3
Book VII: Applications	3
How to Use This Book	3
Icons Used in This Book.....	4
Getting Started	4

Book 1: Getting Started.....5

Chapter 1: Getting Started Programming a Computer 7

How Computer Programming Works	7
Identifying the problem	8
Defining the steps.....	9
The History of Computer Programming.....	10
Talking to a processor in machine language.....	11
Using assembly language as a shortcut to machine language	12
Hiding the details of a computer with a high-level language.....	14
Combining the best of both worlds with the C programming language.....	14
Pros and cons of programming languages	15
Figuring Out Programming.....	17
Desire beats technical training every time	17
Picking a computer and an operating system	18
Writing programs with an editor	19
Converting source code with an assembler or compiler	21
Translating source code with an interpreter.....	22
Combining a compiler with an interpreter to create p-code	23
Taking the time to understand	23
Getting Started with Programming.....	24
Starting with Windows.....	24
Starting with Mac OS X.....	25
Starting with Linux.....	25
Starting with Java.....	26
Knowing Programming Versus Knowing Programming Language.....	26

Chapter 2: Different Methods for Writing Programs29

Spaghetti Programming without a Plan.....	31
Planning Ahead with Structured Programming.....	32
The three parts of structured programming	33
Top-down programming	35
Making User Interfaces with Event-Driven Programming	38
Designing a user interface.....	40
Writing event handlers	42
Writing your program	43
Organizing a Program with Object-Oriented Programming.....	43
Objects isolate data	45
Objects simplify modifications	46
Designing Programs with Today's Methodology.....	48

Chapter 3: Types of Programming Languages49

Choosing Your First Language.....	50
C the BASIC choices	50
Having second thoughts.....	50
Teaching Languages.....	52
Getting back to BASIC	52
Turtle graphics with Logo	55
The philosophy of Pascal	57
Other teaching languages.....	58
“Curly Bracket” Languages	63
The power of C	63
The efficiency of C	64
The portability of C	65
Adding object-oriented programming with C++	65
True portability with Java	66
Safer programming with C#.....	66
Choosing a curly bracket language	69
Artificial Intelligence Languages	70
Scripting Languages.....	73
Automating a program.....	75
Customizing a program.....	76
Transferring data among multiple programs.....	77
Creating separate applications.....	77
Database Programming Languages.....	78
The dBASE programming language.....	78
Adding database access to traditional languages	80
Customizing database programs.....	80
Comparing Programming Languages.....	81

Chapter 4: Programming Tools83

Choosing a Compiler	84
Defining your needs for a compiler.....	84
Evaluating the technical features of a compiler	86

Finding an Interpreter.....	90
Operating system interpreters	91
Web page interpreters	91
Compiling to a Virtual Machine.....	93
Writing a Program with an Editor	94
Editors.....	94
Fixing a Program with a Debugger	97
Stepping line by line.....	97
Watching variables	100
Saving Time with Third-Party Components.....	102
Optimizing a Program with a Profiler	102
Creating a Help File	103
Installing a Program.....	104
Dissecting Programs with a Disassembler.....	105

Chapter 5: Managing Large Projects with Software Engineering107

Software Engineering Methods.....	108
Designing a program with the waterfall model.....	108
Evolving a program with extreme programming.....	112
Automating Software Engineering with CASE.....	115
Modeling a large project.....	115
Generating code automatically.....	119
Formatting source code automatically.....	120
Tracking revisions in code	121
The Pros and Cons of Software Engineering.....	122

Book 11: Programming Basics 125

Chapter 1: How Programs Work127

Using Keywords as Building Blocks.....	128
Organizing a Program	130
Dividing a Program into Subprograms	132
Dividing a Program into Objects	135
Creating a User Interface.....	138

Chapter 2: Variables, Data Types, and Constants141

Declaring Variables	142
Creating a variable	142
Using Different Data Types	148
Storing Data in a Variable.....	150
Retrieving Data from a Variable.....	151
Using Constant Values	153
Defining the Scope of a Variable.....	154
Handling global variables with care.....	154
Restricting scope to a module.....	156

Isolating variables in a subprogram.....	157
Passing data among subprograms	158
Chapter 3: Manipulating Data	161
Storing Data with the Assignment Operator.....	162
Using Math to Manipulate Numbers.....	162
Organizing equations with operator precedence.....	164
Using built-in math functions.....	165
Manipulating Strings.....	166
Finding Strings with Regular Expressions.....	168
Pattern matching with the single character (.) wildcard	168
Pattern matching for specific characters.....	169
Pattern matching with the multiple character (*) and (+) wildcards	169
Pattern matching with ranges.....	170
Using Comparison Operators	172
Using Boolean Operators	174
Using the Not operator	175
Using the And operator	175
Using the Or operator	176
Using the Xor operator	177
Converting Data Types	178
Chapter 4: Making Decisions by Branching	181
Picking One Choice with the IF-THEN Statement.....	182
Picking Two Choices with the IF-THEN-ELSE Statement.....	184
Picking Three or More Choices with the IF-THEN-ELSEIF Statement.....	185
Checking a condition for each set of commands	186
Offering three or more choices	187
Playing with Multiple Boolean Operators.....	189
Making Multiple Choices with the SELECT CASE statement	190
The switch statement in C (and similar languages).....	191
Matching multiple values in a SELECT CASE statement.....	193
Checking a range of values.....	194
Comparing values.....	195
Running at least one command with the ELSE statement.....	195
Chapter 5: Repeating Commands by Looping	199
Looping a Fixed Number of Times with the FOR-NEXT Loop.....	200
Using a FOR-NEXT loop variable	201
Counting by a different range	203
Counting by different increments	204
Counting backward	205
Looping Zero or More Times with the WHILE Loop	206
Looping at Least Once with the DO Loop	208

Playing with Nested Loops	209
Prematurely Exiting from a Loop	211
Checking Your Loops	211
Chapter 6: Breaking a Large Program into Subprograms	213
Creating and Using Subprograms.....	214
Creating a subprogram	216
“Calling” a subprogram.....	217
Passing Parameters.....	219
Passing parameters by reference	222
Storing values in a subprogram name	225
Repeating a Subprogram with Recursion.....	227
Chapter 7: Breaking a Large Program into Objects	231
How Object-Oriented Programming Works	232
Encapsulation Isolates Data and Subprograms.....	235
Shielding data inside an object.....	236
Grouping subprograms inside of an object.....	237
Protecting code from other programmers	238
Sharing Code with Inheritance	238
Polymorphism: Modifying Code without Changing Its Name.....	242
Design Patterns	243
Object-Oriented Languages	246
Hybrid languages.....	246
Pure languages.....	247
Disadvantages of object-oriented programming	248
Real-Life Programming Examples.....	249
Defining an object with a class	249
Creating an object from a class	251
Running subprograms stored in an object.....	252
Inheriting an object	253
Using polymorphism to rewrite an inherited subprogram	256
Chapter 8: Reading and Saving Files	259
Storing Data in Text Files	259
Creating a text file	261
Reading a text file	263
Storing Fixed Size Data in Random-Access Files	264
Writing data.....	266
Reading data	267
Storing Varying Size Data in Untyped Files	268
Writing data.....	268
Reading data	269
Using Database Files	271
Structure of a database.....	272
Connecting to a database	273

Chapter 9: Documenting Your Program277

Adding Comments to Source Code	277
Line comments	279
Block comments	279
Describing code and algorithms	282
Documentation	283
Debugging.....	284
Writing Software Documentation	285
Documentation types.....	285
Documentation tools	286
Help files	287

Chapter 10: Principles of User Interface Design289

The Evolution of User Interfaces.....	290
Command-line interface	290
Menus.....	290
Graphical user interface	292
Elements of a User Interface	293
Displaying commands to a user interface.....	293
Giving data to the user interface	296
Showing information back to the user.....	300
Organizing a user interface	301
Designing a User Interface	302
Know the user	303
Hide unusable options	304
Tolerate mistakes	305
Be consistent	306
Focus on the task	306
Make navigation easy.....	307

Book III: Data Structures309**Chapter 1: Structures and Arrays311**

Using Structures	311
Storing data.....	312
Retrieving data.....	313
Using an Array	314
Defining the size	314
Storing data.....	318
Retrieving data.....	318
Working with Resizable Arrays.....	319
BASIC.....	319
C++	320
Working with Multi-Dimensional Arrays	321
Creating a multi-dimensional array	322
Storing and retrieving data	323

Using Structures with Arrays.....	323
Drawbacks of Arrays.....	325
Sizing.....	325
Data types.....	326
Searching and sorting.....	326
Adding and deleting.....	326
Chapter 2: Sets and Linked Lists	329
Using Sets.....	329
Adding (and deleting) data in a set.....	331
Checking for membership.....	332
Manipulating two sets.....	333
Using Linked Lists.....	337
Creating a linked list.....	338
Modifying a linked list.....	339
Creating a double linked list.....	340
Drawbacks of Sets and Linked Lists.....	341
Problems with pointers.....	342
Problems with accessing data.....	342
Chapter 3: Collections and Dictionaries	345
Using a Collection.....	345
Adding data to a collection.....	346
Deleting data from a collection.....	348
Identifying data with keys.....	349
Searching and retrieving data.....	350
Using Dictionaries.....	352
Adding data to a dictionary.....	352
Searching and retrieving data from a dictionary.....	353
Understanding Hash Tables.....	353
Converting keys with a hash function.....	354
Hash function collisions.....	356
Chapter 4: Stacks, Queues, and Deques	361
Using a Stack.....	361
Adding data to a stack.....	362
Removing data from a stack.....	363
Counting and searching a stack.....	365
Using Queues.....	365
Adding data to a queue.....	367
Removing data from a queue.....	368
Counting and searching a queue.....	368
Using Deques.....	370
Chapter 5: Graphs and Trees	375
Understanding Graphs.....	376
Types of graphs.....	377
Uses for graphs.....	378

Creating Trees.....	380
Ordered trees	381
Binary trees.....	382
B-trees	383
Taking Action on Trees.....	384
Traversing a tree	385
Adding new data.....	386
Deleting data	386
Pruning and grafting sub-trees	387

Book IV: Algorithms391

Chapter 1: Sorting Algorithms393

Using Bubble Sort	394
Using Selection Sort.....	396
Using Insertion Sort	397
Using Shell Sort	398
Using Heap Sort.....	400
Using Merge Sort	403
Using Quick Sort.....	405
Comparing Sorting Algorithms.....	406

Chapter 2: Searching Algorithms409

Sequential Search.....	410
Backward or forward searching	411
Block searching	412
Binary searching.....	413
Interpolation searching	414
Using Indexes.....	416
Creating an index.....	416
Clustered and unclustered indexes.....	417
Problems with indexes	418
Adversarial Search.....	418
Depth versus time	419
Alpha-beta pruning.....	420
Looking up a library of good moves	421

Chapter 3: String Searching423

Sequential Text Search	424
The Boyer-Moore algorithm.....	425
The Rabin-Karp algorithm.....	426
The Shift Or algorithm	427
The finite automaton string search algorithm.....	428

Searching with Regular Expressions.....	429
Searching for single character patterns	430
Searching for multiple character patterns	430
Searching for alternate patterns.....	431
Searching Phonetically	431

Chapter 4: Data Compression Algorithms435

Lossless Data Compression Algorithms.....	436
Run-length encoding	436
The Burrows-Wheeler transform algorithm.....	436
Dictionary encoding.....	439
Lossy Data Compression.....	442

Chapter 5: Encryption Algorithms445

The Basics of Encryption	447
Stream ciphers	449
Block ciphers	450
Symmetric/Asymmetric Encryption Algorithms.....	452
Cracking Encryption	455
Brute force attack.....	456
Dictionary attacks	457
Plaintext and ciphertext attacks	458

Book V: Web Programming.....461

Chapter 1: HyperText Markup Language463

The Structure of an HTML Document.....	463
Creating a title.....	464
Creating the body text	464
Aligning text	466
Emphasizing text	467
Adding color.....	467
Changing the font size	468
Adding comments	469
Adding Graphics.....	469
Defining the Background.....	469
Creating Hyperlinks	470
Defining an anchor point.....	470
Linking to an anchor point	471
Making Tables.....	471
Defining a table	471
Defining a table heading.....	472
Creating table rows and data.....	473
Displaying a table caption, header, and footer.....	474

Chapter 2: CSS	477
The Structure of a Stylesheet	477
Creating Style Classes	479
Separating Styles in Files	481
Cascading Stylesheets	482
Chapter 3: JavaScript	485
The Structure of a JavaScript Program	486
Creating Comments	487
Declaring Variables	487
Using Operators	488
Increment and decrement operators	489
Assignment operators	490
Branching Statements	490
Looping Statements	492
Creating Functions	493
Using Arrays	493
Designing User Interfaces	494
Creating dialog boxes	494
Creating windows	496
Chapter 4: PHP	497
The Structure of a PHP Program	497
Creating Comments	498
Declaring Variables	499
Using Operators	500
Increment and decrement operators	501
Assignment operators	502
Branching Statements	502
Looping Statements	504
Creating Functions	505
Using Arrays	506
Creating Objects	507
Chapter 5: Ruby	509
The Structure of a Ruby Program	510
Creating Comments	510
Declaring Variables	511
Using Operators	511
Branching Statements	514
Looping Statements	515
Creating Functions	516
Using Data Structures	517
Creating Objects	518

Book VI: Programming Language Syntax.....521

Chapter 1: C and C++523

The Structure of a C/C++ Program	524
Creating Comments	525
Declaring Variables	525
Declaring string data types	526
Declaring integer data types	526
Declaring floating point data types	527
Declaring Boolean values	528
Using Operators	528
Increment and decrement operators	529
Assignment operators	530
Branching Statements	531
Looping Statements	533
Creating Functions	534
Data Structures	535
Creating a structure	536
Creating enumerated variables	536
Creating an array	537
Using Objects	537

Chapter 2: Java and C#541

The Structure of a Java/C# Program	542
Creating Comments	542
Declaring Variables	543
Declaring string data types	543
Declaring integer data types	544
Declaring floating point data types	545
Declaring Boolean variables	545
Using Operators	545
Increment and decrement operators	546
Assignment operators	547
Branching Statements	548
Looping Statements	551
Creating Functions	552
Data Structures	553
Creating a C# structure	554
Creating an array	554
Creating a Java linked list	555
Creating C## data structures	556
Using Objects	556

Chapter 3: Perl and Python559

The Structure of a Perl/Python Program	560
Creating Comments	561
Defining Variables	561

Using Operators	561
Increment and decrement operators	563
Assignment operators	564
Branching Statements	565
Looping Statements	566
Creating Functions	568
Perl Data Structures	569
Creating a Perl array	569
Creating a Perl hash array	570
Python Data Structures	570
Creating a Python tuple	571
Creating a Python list	571
Creating a Python dictionary	572
Using Objects	572
Chapter 4: Pascal and Delphi	575
The Structure of a Pascal Program	576
Creating Comments	577
Declaring Variables	577
Declaring string data types	577
Declaring integer data types	578
Declaring decimal data types	579
Declaring Boolean values	579
Declaring Constants	579
Using Operators	580
Branching Statements	581
Looping Statements	583
Creating Subprograms and Functions	584
Data Structures	585
Creating a record	585
Creating an array	586
Creating a set	587
Creating Objects	587
Chapter 5: Visual Basic and REALbasic	589
The Structure of a BASIC Program	590
Using windows files	590
Using module files	591
Using class files	591
Creating Comments	592
Declaring Variables	592
Declaring string data types	592
Declaring integer data types	593
Declaring decimal data types	594
Declaring Boolean values	595
Declaring generic values	595

Declaring Constants.....	595
Using Operators	596
Branching Statements	597
Looping Statements	600
Creating Subprograms and Functions	601
Data Structures.....	603
Creating a structure	603
Creating an array.....	603
Creating a collection and a dictionary.....	604
Creating Objects.....	605

Book VII: Applications.....607

Chapter 1: Database Management609

The Basics of Databases.....	609
Free-form databases.....	610
Flat-file databases.....	611
Relational databases	613
Manipulating Data	617
Writing database commands	620
The SQL language.....	620
Data integrity	621
Data mining	622
Database Programming	622

Chapter 2: Bioinformatics625

The Basics of Bioinformatics.....	625
Representing molecules	626
Manipulating molecules in a computer	627
Searching Databases.....	628
Bioinformatics Programming.....	630

Chapter 3: Computer Security633

Stopping Malware.....	634
Viruses	634
Worms	635
Trojan horses	636
Spyware	636
Stopping Hackers	637
Intrusion detection systems	637
Rootkit detectors.....	638
Forensics	639
Secure Computing.....	639
Patching as an afterthought.....	640
Security in coding.....	640
Security by design	641

Chapter 4: Artificial Intelligence	643
Problem Solving	644
Game-playing	645
Expert systems	646
Natural language processing	648
Speech recognition	650
Image recognition	651
Machine Learning	652
Bayesian probability	653
Neural networks	653
Applications in Artificial Intelligence	656
Chapter 5: The Future of Computer Programming	657
Picking a Programming Language	657
Picking an Operating System	658
Cross-Platform Programming	660
The portability of C	660
Cross-platform languages	661
Virtual machines	662
Software as service	663
Rich Internet applications (RIA)	664
Robotics programming	665
The Programming Language of the Future	666
Low-level languages	666
The next generation: C++, Objective-C, C#, and Java	667
REALbasic and Visual Basic	667
The scripting languages	668
The best programming language	668
Index	671

Introduction

If you enjoy using a computer, you may have even more fun learning to control a computer by writing your own programs. To learn how to program a computer, you need to understand three different subjects.

First, you have to understand that computer programming is nothing more than problem solving. Before you even think about writing a program, you need to know what problem you want your program to solve and how it will solve it.

Second, you need to learn the basic ideas behind computer programming in general. Although programming a Windows computer is different from programming a Macintosh or a super computer, the general principles are the same. By learning what these common programming principles are and why they exist, you can learn different ways to tell a computer what to do, step-by-step.

Finally, you also need to learn a specific programming language. A programming language represents just one way to express your ideas in a language that the computer can understand. By combining your knowledge of a programming language with programming principles and the type of problem you want the computer to solve, you can create your own computer programs for fun or profit.

Who Should Buy This Book

If you have any interest in programming but don't know where to start, this book can give you a nudge in the right direction. You won't learn how to write programs in a specific programming language, but you'll learn the basics about computer programming so you'll have no trouble learning more on your own.

If you already know something about programming, this book can still help you learn more by introducing you to the variety of programming languages available and make it easy for you to learn different programming languages quickly. The more you understand the advantages and disadvantages of different programming languages, the better you'll be able to choose the language that's best suited for a particular task.

Whether you're a novice or an intermediate programmer, you'll find this book can work as a tutorial to teach you more and as a reference to help refresh your memory on programming topics you may not normally use

everyday. This book won't turn you into an expert overnight, but it will open the doors to more information about programming than you might have ever known even existed.

How This Book Is Organized

To help you learn about computer programming, this book is divided into several minibooks where each minibook is self-contained so you can read them in any order you wish.

Book I: Getting Started

This first book provides a general overview of computer programming. This is where you'll learn how programming evolved, the latest techniques to make programming faster and easier, how different programming languages work, what type of tools programmers use, and how programmers create large projects. This book gives you a broad understanding of programming so you'll understand how programming solves different types of problems.

Book II: Programming Basics

No matter what type of computer you want to program or which programming language you use, programming follows the same basic principles and this book explains how they work. This is where you'll learn the parts of a typical program and how they work to create a working program. This book also explains the purpose of a user interface and the elements that make up a user interface.

Book III: Data Structures

All programs manipulate data, which means they need a way to store that data temporarily in something called a data structure. Data structures form the heart of any program. Think of data structures as closets or drawers for holding information. Depending on the type of information you need to store and how you need to use it, you may need to use different types of data structures. By understanding the different data structures available and the pros and cons of each, you'll learn the best type of data structure to use for your particular program.

Book IV: Algorithms

An algorithm is just a specific method for solving a problem. There are literally an infinite number of ways to solve any problem, but computer programmers generally run into the same type of problems over and over, which

means they've also discovered the most efficient algorithms to solve for specific tasks. This book introduces different types of algorithms and explains how they work and when you might want to use them.

Book V: Web Programming

Traditional programs run on a computer, but the advent of the Internet has created a new field of programming known as Web programming. Instead of running on a single computer, Web programs are stored on one computer (called a server) but run and displayed on a second computer (called a client). If you've ever ordered anything off the Internet, you've used a program designed to run on Web pages. Learning Web programming isn't hard, but it often requires using different types of languages and techniques, which is what you'll learn about in this book.

Book VI: Programming Language Syntax

This book explains the basic features of programming languages and then shows how different programming languages implement those features, which is known as the language syntax. Knowing the syntax of different languages can show you the flaws and benefits of each language so you'll know when to use a particular language over another. If you regularly need to program in different languages, this book makes a handy reference to show you how to accomplish a specific task in different languages along with showing you unusual features that appear in one programming language but not in another one.

Book VII: Applications

Knowing how to program a computer is nice, but ultimately useless if you don't know how to apply your programming skills to doing something productive, so this book explains several fields of computer programming. By reading this book, you can learn what types of programming appeals to you. By combining your knowledge of programming with a particular field of interest, you can carve a niche for yourself in any field you choose.

How to Use This Book

You can use this book as a tutorial or a reference. Although you can just flip through this book to find the information you need, programming novices should start with Book I before tackling any other books. After you understand the basics of programming from Book I, then you can freely jump around to read only the information that interests you.

Icons Used in This Book

Icons highlight important or useful information that you might want to know about.



This highlights information that can save you time or make it easier for you to do something.



This icon emphasizes information that can be helpful, though not crucial, when writing a program.



Look out! This icon highlights something dangerous that you need to avoid before making an irreversible mistake that could make you curse your computer forever.



This icon highlights interesting technical information that you can safely ignore, but which might provide additional background about programming a computer.

Getting Started

The best way to learn anything is to jump right in and not be afraid to make mistakes. To learn programming, you'll need access to a computer and a way to write programs in a particular programming language. You won't learn specific programming techniques in this book. Instead, you'll learn a lot about a wide variety of programming topics so you'll be in a better position to decide what you need to learn next.

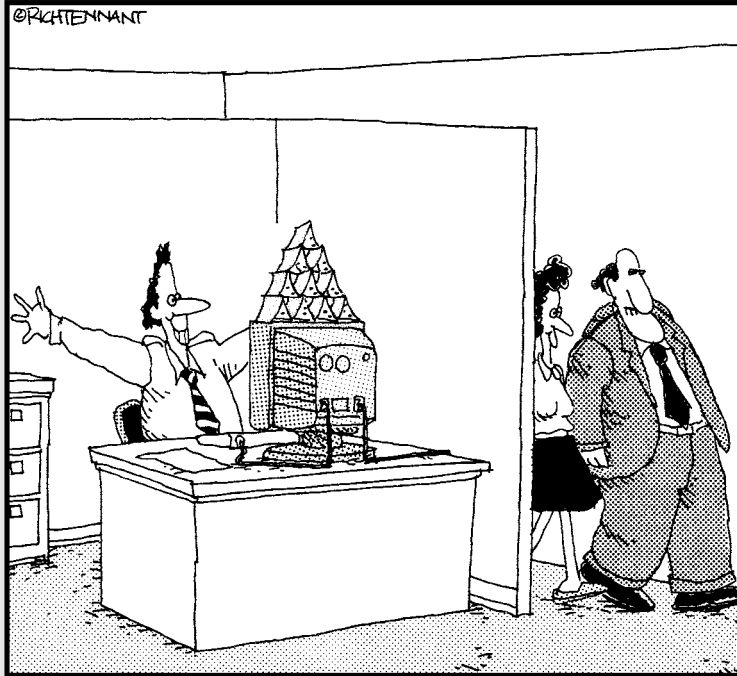
Programming is more than learning a particular programming language or even knowing how to program a particular type of computer. Basically, programming is about tackling difficult problems and breaking them down into smaller problems until you solve one big problem. If you like the idea of solving problems, then you'll find this may be the perfect book to introduce you to the wonderful world of computer programming.

Book I

Getting Started

The 5th Wave

By Rich Tennant



"Why, of course. I'd be very interested in seeing this new milestone in the project."

Contents at a Glance

Chapter 1: Getting Started Programming a Computer	7
Chapter 2: Different Methods for Writing Programs.....	29
Chapter 3: Types of Programming Languages	49
Chapter 4: Programming Tools	83
Chapter 5: Managing Large Projects with Software Engineering.....	107