

ROBOT PROGRAMMER'S BONANZA

JOHN BLANKENSHIP

SAMUEL MISHAL



**New York Chicago San Francisco Lisbon London Madrid
Mexico City Milan New Delhi San Juan Seoul
Singapore Sydney Toronto**

The McGraw-Hill Companies

Library of Congress Cataloging-in-Publication Data

Blankenship, John, date.

Robot programmer's bonanza / John Blankenship, Samuel Mishal.

p. cm.

ISBN 978-0-07-154797-0 (alk. paper)

1. Robotics. I. Mishal, Samuel. II. Title.

TJ211.B565 2008

629.8'9251—dc22

2008010227

Copyright © 2008 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States 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 data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 1 4 3 2 1 0 9 8

ISBN 978-0-07-154797-0

MHID 0-07-154797-5

Printed and bound by RR Donnelley.

This book is printed on acid-free paper.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. To contact a special sales representative, please visit the Contact Us page at www.mhprofessional.com.

Sponsoring Editor

Judy Bass

Production Supervisor

Richard C. Ruzycka

Editorial Supervisor

Stephen M. Smith

Project Manager

Vastavikta Sharma, International
Typesetting and Composition

Copy Editor

Priyanka Sinha, International Typesetting
and Composition

Proofreader

Divya Kapoor, International Typesetting
and Composition

Art Director, Cover

Jeff Weeks

Composition

International Typesetting and Composition

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

ABOUT THE AUTHORS

JOHN BLANKENSHIP taught computer and electronic technology for 33 years at the college level. He has also worked as an engineer and as an independent consultant. He received a B.S. in electrical engineering from Virginia Tech, a masters in electronic engineering technology from Southern Polytechnic State University, and an M.B.A. from Georgia State University. This is his sixth book.

SAMUEL MISHAL is a software engineer and systems analyst. He worked as a consultant for major government departments and businesses around the world. He taught mathematics and computing at the college level. He received a B.S. in electronics engineering technology from DeVry University, a bachelors in computer science from the University of Western Australia, a masters in engineering science from Oxford University, and a masters in structural engineering from Imperial College London.

CONTENTS

Preface	xxiii
Acknowledgments	xxvii
PART 1—BUILDING BLOCKS	1
Chapter 1—Why Simulations	3
1.1 What Is RobotBASIC?	3
1.2 Flight Simulators	4
1.3 Comparing RobotBASIC with Other Simulators	4
1.4 Developing Robot Behaviors	5
1.5 Simulation Can Improve Hardware Choices	6
1.6 Robots Are Not Just Hardware	6
1.7 RobotBASIC Teaches Programming	7
1.8 Summary	7
Chapter 2—Introduction to RobotBASIC	9
2.1 Running RobotBASIC	10
2.2 The RobotBASIC IDE	10
2.2.1 The Editor Screen	10
2.2.2 The Terminal Screen	11
2.2.3 The Help Screen	11
2.3 Creating, Running, and Saving a Program	12
2.4 The Robot Simulator	13
2.4.1 Initializing the Robot	13
2.4.2 Animating the Robot	14
2.4.3 Moving Around Obstacles	16
2.5 Summary	17
2.6 Exercises	17

Chapter 3—RobotBASIC Sensors	19
3.1 Some Programming Constructs	19
3.1.1 Comments	20
3.1.2 Conditional Statements	20
3.1.3 Comparison Operators	21
3.1.4 Loops	21
3.1.5 Binary Numbers	21
3.2 Avoiding Collisions Using Bumpers	22
3.2.1 Bumper Sensor	23
3.2.2 Avoiding Collisions	24
3.2.3 Improving Efficiency	25
3.2.4 Making Better Decisions	25
3.3 Other Sensors for Object Detection	26
3.3.1 Infrared Sensors	26
3.3.2 Ultrasonic and Infrared Ranging	27
3.3.3 Robot Vision	27
3.3.4 Beacon Detection	28
3.3.5 Customizable Sensors	28
3.4 Other Instruments	28
3.4.1 Compass	29
3.4.2 Global Positioning	29
3.4.3 Battery Charge Level	30
3.5 Summary	30
3.6 Exercises	30
Chapter 4—Remote Control Algorithms	33
4.1 Some Programming Constructs	34
4.1.1 Variables	34
4.1.2 The Keyboard	35
4.1.3 The Mouse	35
4.1.4 Output to the Screen	36
4.1.5 Loops	36
4.1.6 Functions	38
4.2 Simple Remote Control	38
4.2.1 First Style of Remote Control	38
4.2.2 Second Style of Remote Control	40
4.3 Complex Remote Control	41
4.3.1 The Mathematics	41
4.3.2 The Pen	44
4.3.3 Subroutines	44

4.4 Remote Controlled Test Bench	47
4.5 Summary	47
4.6 Exercises	49
Chapter 5—Random Roaming	51
5.1 What Is Random Roaming?	52
5.2 Some Programming Constructs	52
5.2.1 Labels and Subroutines	53
5.2.2 Commands	54
5.2.3 Operators	54
5.3 Adding Objects to the Roaming Environment	55
5.3.1 <i>DrawObjects</i> Subroutines	57
5.3.2 <i>RoamAround</i> Subroutines	58
5.4 More Intelligent Roaming	59
5.4.1 Using Sensory Information More Effectively	59
5.5 Improved Obstacle Avoidance	61
5.5.1 A First Improvement	61
5.5.2 A Second Improvement	61
5.5.3 Further Improvements	62
5.6 Summary	63
5.7 Exercises	63
Chapter 6—Debugging	65
6.1 Before You Program	66
6.2 Plan Plan Plan	66
6.3 Debugging Philosophy	67
6.3.1 Isolating the Fault	67
6.3.2 Locating the Fault	67
6.3.3 Correcting the Problem	68
6.3.4 Patience Patience Patience	68
6.4 Debugging with RobotBASIC	68
6.4.1 The Debug Command	69
6.4.2 Stepping Through a Program	70
6.4.3 Viewing the Infrared Beams	71
6.4.4 Viewing Bumper LEDs	71
6.5 Summary	71
6.6 Exercises	72

PART 2—DEVELOPING A TOOLBOX OF BEHAVIORS	73
Chapter 7—Following a Line	75
7.1 The Base Program	76
7.2 An Initial Algorithm	77
7.2.1 Reading the Line Sensors	79
7.2.2 A First Attempt	79
7.2.3 An Improvement	80
7.3 Sharp Turns Cause a Problem	80
7.3.1 Possible Solutions	80
7.3.2 A First Strategy	81
7.3.3 A Second Strategy	82
7.3.4 Very Sharp Turns	82
7.4 Random Roaming with Line-Following (Racetrack)	83
7.4.1 The <i>RoamAround</i> Subroutine	85
7.4.2 The <i>InitializeRobot</i> Subroutine	85
7.4.3 The Data Statement and <i>mPolygon</i> Command	85
7.5 Summary	87
7.6 Exercises	88
Chapter 8—Following a Wall	89
8.1 Constructing a Wall	89
8.2 A Basic Algorithm	91
8.2.1 Problems with the Basic Algorithm	91
8.2.2 Improving the Algorithm	92
8.2.3 Using the Bumpers	93
8.3 Staying Close on Sharp Corners	94
8.3.1 Initial Algorithm	94
8.3.2 Finding the Problem	95
8.3.3 Solving the Problem	96
8.4 A Different Approach	96
8.5 Summary	100
8.6 Exercises	101
Chapter 9—Avoiding Drop Offs and Restricted Areas	103
9.1 Good Robot	104
9.1.1 An Initial Algorithm	104
9.1.2 Improving the Algorithm	105
9.1.3 A Better Algorithm	106
9.2 Cliff Hanger	108

9.3 GPS Confinement	112
9.3.1 The Specifications	112
9.3.2 Main Program	113
9.3.3 <i>RoamAround</i> Subroutine	113
9.3.4 <i>DrawBoundary</i> Subroutine	115
9.3.5 <i>TestViolation</i> Subroutine	115
9.4 Summary	119
9.5 Exercises	120
Chapter 10—Vector Graphics Robot	121
10.1 DrawBot	122
10.1.1 Drawing Circles	122
10.1.2 Drawing Rectangles	123
10.1.3 Drawing Triangles	124
10.1.4 Drawing any Shape	126
10.2 ABC Robot	128
10.2.1 The Specifications	128
10.3 Summary	134
10.4 Exercises	134
PART 3—COMPLEX COMPOUND BEHAVIORS	137
Chapter 11—Mowing and Sweeping Robot	139
11.1 Sweeper Robot	139
11.1.1 The Base Program	140
11.1.2 A First Attempt	140
11.1.3 An Improvement	140
11.1.4 Further Improvements	144
11.2 Mowing Robot	146
11.2.1 The Specifications	147
11.2.2 The Program	147
11.2.3 A Shortcoming	153
11.3 Further Thoughts	154
11.3.1 Considering the Batteries	154
11.3.2 Limited Coverage Around Obstacles	154
11.3.3 Using GPS Grids	155
11.3.4 A Reality Check	155
11.4 Summary	155
11.5 Exercises	156

Chapter 12—Locating a Goal	157
12.1 Using a Beacon	158
12.1.1 The Algorithm	158
12.1.2 The Main Program	159
12.1.3 Creating a Cluttered Room	159
12.1.4 Facing the Beacon	160
12.1.5 Moving Toward the Beacon	160
12.1.6 Going Around an Obstacle	161
12.1.7 Determining If the Beacon Is Found	162
12.1.8 A Potential Problem	162
12.2 Using a Beacon and Camera	163
12.3 Using a GPS and Compass	165
12.4 Summary	166
12.5 Exercises	166
Chapter 13—Charging the Battery	169
13.1 The Robot's Battery	170
13.2 Real-World Charging	171
13.2.1 Finding the Station	172
13.2.2 The Charging Station	172
13.2.3 Ensuring a Proper Approach Angle	172
13.3 The Simulation	172
13.3.1 Subroutines Hierarchy Chart	173
13.3.2 The Program	174
13.4 Summary	179
13.5 Exercises	179
Chapter 14—Negotiating a Maze	181
14.1 A Random Solution	182
14.1.1 The Program	182
14.1.2 Observations	188
14.2 A Directed Random Solution	189
14.3 A Minimized Randomness Solution	190
14.3.1 A Corridor Maze	191
14.3.2 The Program	191
14.3.3 Generating the Maze	191
14.3.4 Solving the Maze	191
14.3.5 Renegotiating the Maze	198
14.3.6 Embedded Debug Commands	198

14.4 A Mapped Solution	198
14.4.1 Mapping the Maze	199
14.4.2 The Program	200
14.4.3 Creating the Map's Graph	200
14.4.4 Solving the Maze	200
14.4.5 Finding a Path	204
14.4.6 The Optimal Path	204
14.5 Final Thoughts	206
14.6 Summary	206
14.7 Exercises	207
Chapter 15—Negotiating a Home or Office	209
15.1 The Design Process	210
15.2 An Office Messenger Robot	210
15.2.1 The Office Specifications	211
15.2.2 The Main Program and Subroutines Hierarchy Chart	211
15.2.3 The User Interface	212
15.2.4 Drawing the Office and Placing the Robot	216
15.2.5 Mapping the Office	219
15.2.6 Waiting for a Command	224
15.2.7 Executing the Command	225
15.2.8 Recharging the Battery	227
15.3 A Reality Check	228
15.3.1 Counteracting Motor Slip with a GPS and Compass	229
15.3.2 No GPS or Compass (Slip Is Corrected by Hardware)	230
15.3.3 Resilience Against Slip Using Beacons	235
15.4 Further Thoughts	235
15.5 Summary	237
15.6 Exercises	237
PART 4—GOING FURTHER	241
Chapter 16—True Intelligence: Adaptive Behavior	243
16.1 Adaptive Behavior	244
16.1.1 Adaptive Wall-Following	245
16.1.2 Adaptive Line-Following	245
16.2 How to Define Intelligence?	246
16.2.1 Human Intelligence	246
16.2.2 Intelligence Through Association	247

xviii CONTENTS

16.3	Adaptation Through Association	248
16.3.1	I Feel Pleasure I Feel Pain	248
16.3.2	Environmental Factors	249
16.4	Implementing the Algorithm	249
16.4.1	Developing a Personality	257
16.4.2	Displaying the Robot's Actions	258
16.4.3	Understanding the Code	259
16.5	Summary	261
16.6	Exercises	262
Chapter 17—Relating Simulations to the Real World		263
17.1	A Historical Perspective	264
17.1.1	Early Hobby Robotics	264
17.1.2	Hobby Robotics Today	265
17.1.3	The Paradigm Shift	265
17.2	Constructing a Robot	267
17.2.1	Wheel and Base Assembly	268
17.2.2	Bumper Sensors	269
17.2.3	Infrared Perimeter Sensors	271
17.2.4	Line Sensors	273
17.2.5	Ranging Sensor	274
17.2.6	The Compass	274
17.2.7	The GPS	275
17.2.8	The Camera	275
17.2.9	Beacon Detection	276
17.2.10	Practical Consideration	278
17.3	Controlling the Real Robot	279
17.3.1	Control by a Microcontroller	280
17.3.2	Control by an Onboard PC	286
17.3.3	Control by a Remote PC Wirelessly	288
17.3.4	Control by a Remote PC Wirelessly Using an Inbuilt Protocol	290
17.4	Resources	296
17.5	Summary	397
Chapter 18—Contests with RobotBASIC		299
18.1	RobotBASIC Based Contests	299
18.2	Types of Contests	300
18.3	Scoring a Contest	301

18.3.1 Scoring with the Points System	301
18.3.2 Scoring with the Battery	302
18.3.3 Scoring with the Quality of Code	302
18.4 Constructing Contest Environments	302
18.5 Summary	303
18.6 Suggested Activities	303
Chapter 19—RobotBASIC in the Classroom	305
19.1 RobotBASIC within the Learning Process	306
19.2 RobotBASIC as a Motivator	307
19.3 RobotBASIC within the Teaching Process	307
19.4 RobotBASIC at Every Level of Education	308
19.4.1 Grade School	308
19.4.2 Middle School	308
19.4.3 High School	308
19.4.4 College Level	309
19.5 Summary	309
19.6 Suggested Teaching Tasks	310
19.6.1 Grade School	310
19.6.2 Middle School	310
19.6.3 High School	310
19.6.4 College Students	310
PART 5—APPENDICES	311
Appendix A—The RobotBASIC IDE	313
A.1 The Editor Screen	313
A.2 The Terminal Screen	315
A.3 The Help Screen	317
A.4 The Debugger Screen	317
Appendix B—The RobotBASIC Language	319
B.1 Statements	319
B.2 Comments	320
B.3 Assignment Statements	321
B.4 Command Statements	322
B.5 Labels	322
B.6 Flow-Control Statements	323

xx CONTENTS

B.7 Expressions	324
B.7.1 Numbers	324
B.7.2 Strings	325
B.7.3 Simple Variables	325
B.7.4 Arrays	326
B.7.5 Operators	327
B.7.6 Constants	332
B.7.7 Functions	333
Appendix C—Commands, Functions, and Other Details	335
C.1 Labels	336
C.1.1 Alpha-Numerical Style 1	336
C.1.2 Alpha-Numerical Style 2	336
C.1.3 Numerical Style	337
C.2 Assignment Statement	337
C.3 Expressions	338
C.4 Strings	338
C.5 Variables	339
C.6 Flow-Control Statements	339
C.6.1 If-Then Statement	339
C.6.2 If-Elseif Statement	340
C.6.3 For-Next Loop	340
C.6.4 Repeat-Until Loop	341
C.6.5 While-Wend Loop	342
C.6.6 Break Statement	342
C.6.7 Continue Statement	342
C.6.8 Case Construct	342
C.6.9 GoSub Statement	343
C.6.10 OnError Statemet	344
C.6.11 End Command	344
C.6.12 Goto Statement	344
C.7 Command Statements	345
C.7.1 Input and Output Commands	345
C.7.2 Screen and Graphics Commands	350
C.7.3 Array Commands	355
C.7.4 Array Math Commands	357
C.7.5 Other Commands	359
C.7.6 DrawShape Details	360

C.8 Functions	361
C.8.1 Trigonometric Functions	361
C.8.2 Cartesian to Polar Functions	362
C.8.3 Polar to Cartesian Functions	362
C.8.4 Logarithmic and Exponential Functions	362
C.8.5 Sign Conversion Functions	363
C.8.6 Float to Integer Conversion Functions	363
C.8.7 Number and String Conversion Functions	363
C.8.8 String Manipulation Functions	364
C.8.9 Time and Date Functions	365
C.8.10 Probability Functions	366
C.8.11 Statistical Functions	366
C.8.12 Array Functions	367
C.8.13 Other Functions	369
C.8.14 Formatting Codes and Logic	371
C.9 The Robot Simulator Commands and Functions	372
C.9.1 General Information	372
C.9.2 Simulator Commands	373
C.9.3 Simulator Functions	376
C.9.4 Simulator Commands Listed Alphabetically	379
C.9.5 Simulator Functions Listed Alphabetically	379
C.10 Commands and Functions Listed Alphabetically	380
C.10.1 Commands	380
C.10.2 Functions	381
Appendix D—Ports and Serial Input/Output	383
D.1 General Information	383
D.2 Serial I/O Commands	384
D.3 Parallel Ports I/O Commands	385
D.4 Virtual Parallel Port I/O Protocol	386
D.5 General Ports I/O Commands	387
D.6 Robot Simulator Serial I/O Protocol	387
Index	391

PREFACE

The field of hobby robotics has many parallels to personal computing. If you wanted to own a computer in the 1970s, you had to build it yourself. Less than a decade later, you could buy a fully assembled computer and people quickly discovered that programming a computer led to far more enjoyment, satisfaction, and productivity than constructing one.

In the 1980s robot hobbyists spent most of their time building robots from wood and sheet metal. They powered their creations with surplus parts like windshield wiper motors salvaged from car junkyards. So much time was spent in the construction phase that minimal thought was given to the electronic aspects of the project—many of the early robots were controlled with doorbell buttons and relays.

As the personal computer became more powerful a more sophisticated robotics hobbyist began to evolve. They learned more about electronics and started building crude sensors and motor control circuitry that, along with a personal computer, gave their robots, at least, the potential to interact with their environments. These new hobbyists renewed the dream that intelligent robots could actually be built. Unfortunately, most of the people interested in robotics still lacked the required electronics skills and knowledge.

In the years that followed, many books and magazines were published that promised to help robot enthusiasts create circuitry to give their robots more intelligence. However, often, due to complexity and lack of experience, many people had trouble duplicating the authors' works.

Despite all these difficulties, the desire to build personal robots did not diminish. New companies emerged offering robot kits that required minimal experience to build and actuate. These early kits were not programmable, and thus did not satisfy the hobbyists' desire to create intelligent machines. Nowadays there are many companies that offer sophisticated sensors and embedded computers that make it possible to build intelligent, capable and useful robots.

Today, you can buy electronic compasses, ultrasonic rangefinders, GPS systems, infrared perimeter sensors, line and drop-off detectors, color detectors, electronic accelerometers, and even cameras. Reasonable knowledge and often a lot of time are still required to interface these devices to a robot's microcontroller, but the abundance of manuals and books make details available to any hobbyist willing to expend the effort. With sophisticated hardware available to everyone, hobby robotics is now able to turn its attention to programming, finally making it possible to create truly intelligent machines.

Considering these developments, it is easy to feel like all the hard work has been done, when in fact, the real work is just beginning. Remember, personal computers were just a curiosity until the emphasis shifted from building them to programming

xxiv PREFACE

them. This *paradigm shift* enabled innovative hobbyists and entrepreneurs to create word processors, spreadsheets, and graphical user interfaces (GUIs) that changed the world. The world of hobby robotics is now entering such an era. Today's robot enthusiasts no longer need a degree in electronics and a machine shop in their garage to create robots that are ready to be programmed. They do, however, need to understand programming, because it is software that truly creates a useful robot.

Sophisticated kits and fully assembled robots are available from many vendors. Numerous companies offer off-the-shelf hardware modules that enable a typical hobbyist to *assemble* a custom robot with capabilities that were only a dream a few years ago. A hobbyist that understands the concepts of robot programming can use these new platforms to create the projects robot builders have been seeking for years.

Unfortunately learning to program a robot can be very frustrating, even if you have the appropriate hardware. Sensors often need adjusting and realigning and batteries always seem to need recharging. When the robot fails to respond properly you run the risk of damaging it or even your home or furniture. Because you can't *see why* the robot is failing, the task of debugging the code can often be exasperating. With the world of robotics entering its new era, there has to be a better way for hobbyists to learn about *programming* their machines.

This book is aimed at the new hobbyist who is interested in *programming* robots. Today there are numerous microcontrollers that can be used to control robots. These controllers can be programmed using a variety of programming languages (Assembly, C, BASIC, and others). This lack of homogeneity in hardware and software tools make it hard to learn how to program a robot, even if you have previous programming experience.

In reality, the details of the implementation using a specific combination of software and hardware are of secondary concern. What is important in programming a robot to do useful tasks is the *algorithm* that achieves the desired logic. Once the algorithm is determined it can be *easily translated* into any programming language to work on any appropriate microcontroller.

RobotBASIC is a full-featured, interpreted programming language with an integrated robot simulator that can be used to *prototype* projects. The simulator allows you to research various combinations of sensors and environments. You can change the types and arrangements of sensors in seconds, making it possible to experiment with numerous software ideas. You can test your algorithms in environments that would be impractical to create in real life.

The simulated mobile robot is two-dimensional, but programming it lets you learn how to use all the sensors you would expect to find installed on robots costing hundreds if not thousands of dollars. And you will soon discover that programming the simulation is so much like programming the real thing (less all the frustrating aspects) that you will soon forget it is just a circle moving on your screen.

RobotBASIC has capabilities far beyond the robot simulator. It is a powerful programming language with functions that support graphics, animation, advanced mathematics, and access to everything from I/O ports to Bluetooth communication so that you can even use it to control a real-world robot if you choose. When you learn about robot programming with RobotBASIC you won't have to spend months building a robot. You will be able to start programming immediately and never have to worry

about charging batteries or damaging furniture, although you can simulate those events too.

The book is divided into four parts. Part 1 explores the advantages of using a simulator and teaches how to use the simulated robot and its sensors. It also introduces the RobotBASIC language and programming concepts in general. By the time you finish Part 1, you will be able to write and debug simple programs that move the robot around a simulated environment while avoiding objects that block its path.

Part 2 examines everything you typically find hobbyists doing at robot clubs. You will learn ways to make the robot follow a line on the floor, hug a wall, or stay away from a drop-off such as a stairway. All of these topics (and more) are examined with simple easy-to-understand approaches. The simulation is then used to expose problems and deficiencies with the initial approaches. New and better algorithms are then developed and explained. Learning about robotics using this building blocks approach can be very motivational because it is exciting and relevant. As you proceed through the book you will gain more knowledge about programming and problem solving principles. This makes RobotBASIC an ideal first language for teaching students about programming, mathematics, logical thinking, and robotics.

The chapters in Part 3 combine the behaviors developed in Part 2 into compound complex behaviors, that enable the robot to solve real-world projects such as charging the robot's battery, mowing a lawn, solving a maze, locating a goal, and negotiating a home or office environment. As in Part 2, the projects are first explored with simple approaches before introducing more complex concepts. The advanced reader will find this part of the book interesting because many behaviors are evolved using mathematics and computer science topics.

Part 4 explores advanced topics such as adaptive behavior and how RobotBASIC programs can be used to control real-world robots using wireless links. Additionally, ideas are forwarded for why RobotBASIC can be useful in robotic contests and as a teaching tool in the classroom.

The RobotBASIC program along with all the programs in this book can be downloaded from www.RobotBASIC.com. The language is subject to change as alterations and upgrades are implemented. The help files accessible from the latest IDE will have the most valid up-to-date descriptions of all the functionalities of the language. Make sure to always download the latest version and to consult the help files for any new and modified features. Also make sure to check the site for:

- Updated listings of all the programs in the book.
- Solutions for some of the exercises in the book.
- Any corrections to errors that may have slipped into the book.
- Other information and news.