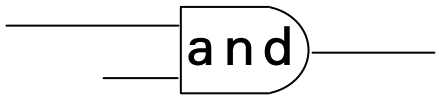
Mathematics
for the Digital Age



Programming
in Python

>>> Second Edition:
with Python 3

Maria Litvin
Phillips Academy, Andover, Massachusetts

Gary Litvin
Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: http://www.skylit.com
e-mail: sales@skylit.com
support@skylit.com

Copyright © 2010 by Maria Litvin, Gary Litvin, and Skylight Publishing

All rights reserved. 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, or otherwise, without the prior written permission of the authors and Skylight Publishing.

Library of Congress Control Number: 2009913596

ISBN 978-0-9824775-8-8

The names of commercially available software and products mentioned in this book are used for identification purposes only and may be trademarks or registered trademarks owned by corporations and other commercial entities. Skylight Publishing and the authors have no affiliation with and disclaim any sponsorship or endorsement by any of these products' manufacturers or trademarks' owners.

1 2 3 4 5 6 7 8 9 10 15 14 13 12 11 10

Printed in the United States of America

Brief Contents

Preface xi
Chapter 1. Sets and Functions 1
Chapter 2. An Introduction to Programming 19
Chapter 3. Variables 37
Chapter 4. Sequences, Sums, Iterations 57
Chapter 5. Number Systems 77
Chapter 6. Boolean Algebra 95
Chapter 7. Digital Circuits and Bitwise Operators 115
Chapter 8. Counting 131
Chapter 9. Strings, Lists, and Files 147
Chapter 10. Parity, Invariants, and Finite Strategy Games 169
Chapter 11. Recurrence Relations and Recursion 189
Chapter 12. Polynomials 205
Chapter 13. Probabilities 221
Chapter 14. Matrices, Sets, and Dictionaries 243
Chapter 15. Graphs 259
Chapter 16. More Graphs 279
Chapter 17. Number Theory and Cryptology 295
Appendices 321
Index 331

Copyright © 2008-2010 by Skylight Publishing

About the Authors

Maria Litvin has taught mathematics and computer science at Phillips Academy in Andover, Massachusetts, since 1987. She is an Advanced Placement Computer Science exam reader and, as a consultant for The College Board, provides AP training for high school computer science teachers. Maria is a recipient of the 1999 Siemens Award for Advanced Placement for Mathematics, Science, and Technology for New England and of the 2003 RadioShack National Teacher Award. Prior to joining Phillips Academy, Maria taught computer science at Boston University. Maria is a co-author of *C++ for You++: An Introduction to Programming and Computer Science*, which became one of the leading high school textbooks for AP Computer Science courses, and *Java Methods A & AB*. Maria is also the author of *Be Prepared for the AP Computer Science Exam in Java*.

Gary Litvin is a co-author of *C++ for You++*, *Java Methods*, and *Be Prepared for the AP Computer Science Exam in Java*. Gary is also the author of *Solutions to 800 Questions in Calculus* and the editor of *Be Prepared for the AP Calculus Exam*. Gary has worked in many areas of software development including artificial intelligence, pattern recognition, computer graphics, and neural networks. As founder of Skylight Software, Inc., he developed SKYLIGHTS/GX, one of the first visual programming tools for C and C++ programmers. Gary led in the development of several state-of-the-art software products including interactive touch screen development tools, OCR and handwritten character recognition systems, and credit card fraud detection software.

Copyright © 2008-2010 by Skylight Publishing

Contents

Preface xi

Chapter 1. Sets and Functions 1

1.1 Prologue 1
1.2 Sets 2
1.3 Ways to Define a Function 6
1.4 Algorithms 11
1.5 Exploring Functions in Python 14
1.6 Review 18

Chapter 2. An Introduction to Programming 19

2.1 Prologue 19
2.2 CPU and Memory 19
2.3 Python Interpreter 24
2.4 Python Code Structure 30
2.5 Review 36

Chapter 3. Variables 37

3.1 Prologue 37
3.2 Variables in Python 37
3.3 Scope of Variables 43
3.4 More About Python Functions 46
3.5 Function Arguments 51
3.6 Review 56

Copyright © 2008-2010 by Skylight Publishing

Chapter 4. Sequences, Sums, Iterations **57**

4.1 Prologue 57
4.2 Arithmetic and Geometric Sequences 58
4.3 Sums 61
4.4 Infinite Sums 63
4.5 Iterations in Python 68
4.6 Review 76

Chapter 5. Number Systems **77**

5.1 Prologue 77
5.2 Positional Number Systems 78
5.3 The Binary, Octal, and Hexadecimal Systems 81
5.4 Representation of Numbers in Computers 87
5.5 Irrational Numbers 90
5.6 Review 94

Chapter 6. Boolean Algebra **95**

6.1 Prologue 95
6.2 Operations on Propositions 96
6.3 Predicates and Sets 99
6.4 `if-else` Statements in Python 105
6.5 Review 113

Chapter 7. Digital Circuits and Bitwise Operators **115**

7.1 Prologue 115
7.2 Gates 118
7.3 Bitwise Logical Operators 123
7.4 Review 130

Copyright © 2008-2010 by Skylight Publishing

Chapter 8. Counting 131

8.1 Prologue 131
8.2 The Multiplication Rule 131
8.3 Permutations 134
8.4 Using Division 137
8.5 Combinations 140
8.6 Using Addition and Subtraction 143
8.7 Review 146

Chapter 9. Strings, Lists, and Files 147

9.1 Prologue 147
9.2 Strings 148
9.3 Lists and Tuples 154
9.4 Files 160
9.5 Review 168

Chapter 10. Parity, Invariants, and Finite Strategy Games 169

10.1 Prologue 169
10.2 Parity and Checksums 170
10.3 Invariants 175
10.4 Finite Strategy Games 179
10.5 Review 188

Chapter 11. Recurrence Relations and Recursion 189

11.1 Prologue 189
11.2 Recurrence Relations 189
11.3 Recursion in Programs 193
11.4 Mathematical Induction 200
11.5 Review 204

Copyright © 2008-2010 by Skylight Publishing

Chapter 12. Polynomials 205

12.1 Prologue 205
12.2 Addition and Subtraction 206
12.3 Multiplication, Division, and Roots 210
12.4 Binomial Coefficients 215
12.5 Review 220

Chapter 13. Probabilities 221

13.1 Prologue 221
13.2 Calculating Probabilities by Counting 222
13.3 More Probabilities by Counting 227
13.4 Multiplication, Addition, and Subtraction 231
13.5 Pseudorandom Numbers 236
13.6 Review 242

Chapter 14. Matrices, Sets, and Dictionaries 243

14.1 Prologue 243
14.2 Tables and Matrices 244
14.3 Sets 250
14.4 Dictionaries 254
14.5 Review 258

Chapter 15. Graphs 259

15.1 Prologue 259
15.2 Types of Graphs 262
15.3 Isomorphism of Graphs 266
15.4 Degree of Vertex 269
15.5 Directed and Weighted Graphs 273
15.6 Review 278

Copyright © 2008-2010 by Skylight Publishing

Chapter 16. More Graphs 279

16.1 Prologue 279
16.2 Adjacency Matrices 279
16.3 Coloring Maps 283
16.4 The Four Color Theorem 287
16.5 Review 294

Chapter 17. Number Theory and Cryptology 295

17.1 Prologue 295
17.2 Euclid's Algorithm 295
17.3 The Fundamental Theorem of Arithmetic 301
17.4 Arithmetic of Remainders 305
17.5 Number Theory in Cryptology 312
17.6 Review 320

Appendices 321

Appendix A. Getting Started with Python 321
Appendix B. Selected Functions: Built-In, `Math`, and `Random` 322
Appendix C. String Operations and Methods 323
Appendix D. List, Set, and Dictionary Operations and Methods 327

Index 331

Copyright © 2008-2010 by Skylight Publishing

How to Use This Book

The *Math and Python* companion web site —

`http://www.skylit.com/python`

— is an integral part of this book. It contains downloadable student files for exercises, assembled together in what we call *Student Disk*. Also on the book's web site are some of the appendices, links, errata, supplemental papers and syllabi, and technical support information for teachers.

`PY` refers to the *Math and Python Student Disk*. For example, "The file of words, `words.txt`, is provided in `PY\Ch09`" means the `words.txt` file is located in the `Ch09` subfolder in your Student Disk folder.

This icon draws your attention to a hands-on exploration of an example.

"Parentheses" like these, in the margin, mark supplementary material intended for a more inquisitive reader. This material either gives a glimpse of things to come in subsequent chapters or adds technical details.

1.▪, **2.**◆ In exercises, a square indicates an "intermediate" question that may require more thought or work than an ordinary question or exercise. A diamond indicates an "advanced" question that could be treacherous or lead to unexplored territory or take a lot of work.

✓ A checkmark at the end of a question in exercises means that the answer or a solution is included on your student disk. We have included answers and solutions to about one-third of the exercises. They can be found in `PY\SolutionsToExercises` (click on `index.html`).

Teacher Disk, which contains complete solutions to all the exercises and labs, is available for downloading free of charge to teachers who use this book as a textbook in their school. Go to `skylit.com/python` and click on the "Teachers' Room" link for details.

Copyright © 2008-2010 by Skylight Publishing

Preface

"So, is this a math book or a computer programming book?" This is probably the first question on the impatient reader's mind. But why should it be? It is a librarian's dilemma: Does it go on the math shelf or on the computer shelf? There is a simple solution: put a copy on each.

The purpose of this book is to teach a particular way of thinking — precision thinking — and how to solve problems that require this way of thinking. Both mathematics and computer programming nourish the ability to think with precision and to solve problems that call for exact solutions.

Mathematics teaches us to appreciate the beauty of a rigorous argument. In the long run, this is more valuable than a lesson on solving today's practical problems. Still, mathematics does not exist in a vacuum — its abstractions are rooted in practical knowledge accumulated over centuries. The teaching of mathematics draws on examples and analogies from the world around us. At least, it should. However, the world around us is changing more and more rapidly. In the past 50 or 60 years, our world has changed dramatically: it has gone digital. This change is so profound that it is sometimes hard to fully comprehend. Is that why the change remains largely ignored in our K-12 math curricula? We need to start filling the gap.

If we could build a time machine and bring Euclid over for a visit, he would find it comforting amid the chaos of modern technologies that geometry familiar to him is still taught in schools. Old rivals Newton and Leibniz would both find great satisfaction in the fact that tens of thousands of 11th and 12th graders are learning how to take derivatives and use integrals. But George Boole, a visitor from the more recent past, would have to search through dozens of school textbooks before he could find his algebra of propositions mentioned even in passing, despite the fact that his name is immortalized in every modern computer programming language. As for John von Neumann, a brilliant mathematician and one of the fathers of computer technology... well, with his usual optimism he would predict that within 20 years or so, every elementary school student will learn about the AND, OR, and NOT gates. And why not?

In this book we have collected some of the easier mathematical topics that are relevant to the digital world. Many of these topics are often bundled together in

Copyright © 2008-2010 by Skylight Publishing

freshman college courses under the name *discrete mathematics*. Discrete mathematics has become a euphemism for all elementary mathematics that is relevant today but neglected in standard middle and high school algebra, precalculus, and calculus courses. In the 1970s, Donald Knuth and his colleagues at Stanford coined the phrase "concrete mathematics" — a blend of CONtinuous and disCRETE mathematics (and also solid and not too abstract) — to describe the course Knuth taught at Stanford. Later, *Concrete Mathematics* became the title of their delightful book.[1] As they explain in the preface, Knuth "had found that there were mathematical tools missing from his repertoire; the mathematics he needed for a thorough, well grounded understanding of computer programs was quite different from what he'd learned as a mathematics major in college."

We believe that starting in college is too late. Many concepts are completely accessible to middle and high school students. And there is also another side to the relationship: just as mathematics helps achieve a deeper understanding of computer programs, some hands-on experience with computer programming helps make mathematics more tangible, familiar, and easier to grasp.

So, if you are interested mainly in computers, we hope this book will make you a better computer programmer. If you are more interested in math, you will have ample opportunities to solve interesting problems and model some of them in computer programs. You will become familiar with fun areas of mathematics that are usually kept from middle and high school students for no obvious reason; you will learn to solve real problems (that is, problems that you don't already know how to "solve" ahead of time); you will learn the power of mathematical reasoning and proof. As a bonus, you will acquire the practical skill of programming in Python, a popular commercial programming language.

We chose Python for several reasons. First, Python gives you a chance to experiment with the language in an interactive setting with immediate feedback. Second, Python's syntax is not too complicated. Third, Python has simple but powerful features for working with lists and "dictionaries" (maps). Finally, Python is easy to install and get started with, and it's free. Of course, there are other programming languages that have similar properties and would meet our needs. In the end, it is not any particular programming language that matters, but the ability to think with precision about both mathematical facts and computer programs.

Copyright © 2008-2010 by Skylight Publishing

[1] Ronald L. Graham, Donald E. Knuth, Oren Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, Second Edition, Addison-Wesley, 1998.

❖ ❖ ❖

We are very grateful to Dr. J. Adrian Zimmer of the Oklahoma School of Science and Mathematics for sharing his ideas about teaching math and Python. Adrian read a draft of this book and made valuable suggestions and corrections.

We thank Kenneth S. Oliver (formerly of Amity Regional High School in Woodbridge, Connecticut) for taking the time to read a draft very thoroughly, pointing out mistakes, and helping to clarify some explanations.

We are very grateful to Prof. Duncan A. Buell, Chair of the Department of Computer Science and Engineering at University of South Carolina in Columbia, for reading a draft and suggesting improvements and corrections, especially for the Number Theory and Cryptology chapter (Chapter 17).

Our thanks to Benjamin Niedzielski, Phillips Academy '08, for writing solutions to many exercises.

❖ ❖ ❖

In the second edition, Python code in all examples and exercises has been converted to Python 3. According to Python developers,

> Python 3.0, also known as 'Python 3000' or 'Py3K', is the first ever intentionally backwards-incompatible Python release. There are more changes than in a typical release, and more that are important for all Python users. Nevertheless, after digesting the changes, you'll find that Python really hasn't changed all that much – by and large, we're mostly fixing well-known annoyances and warts, and removing a lot of old cruft.[*]

We have inserted the "Parity, Invariants, and Finite Strategy Games" chapter after Chapter 9 and split the Graphs chapter into two, adding sections on graph coloring and the Four Color Theorem.

Copyright © 2008-2010 by Skylight Publishing

[*] See http://docs.python.org/3.1/whatsnew/3.0.html.

Copyright © 2008-2010 by Skylight Publishing