

A close-up photograph of a bass guitar body, showing the f-hole, two humbucker pickups, and the bridge area. The wood is a warm, reddish-brown color. The background is black.

Making Music with Computers

Creative Programming in Python

Bill Manaris
Andrew R. Brown

 **CRC Press**
Taylor & Francis Group

A CHAPMAN & HALL BOOK

Making Music with Computers

Creative Programming in Python

CHAPMAN & HALL/CRC TEXTBOOKS IN COMPUTING

Series Editors

John Impagliazzo

Professor Emeritus, Hofstra University

Andrew McGettrick

Department of Computer
and Information Sciences
University of Strathclyde

Aims and Scope

This series covers traditional areas of computing, as well as related technical areas, such as software engineering, artificial intelligence, computer engineering, information systems, and information technology. The series will accommodate textbooks for undergraduate and graduate students, generally adhering to worldwide curriculum standards from professional societies. The editors wish to encourage new and imaginative ideas and proposals, and are keen to help and encourage new authors. The editors welcome proposals that: provide groundbreaking and imaginative perspectives on aspects of computing; present topics in a new and exciting context; open up opportunities for emerging areas, such as multi-media, security, and mobile systems; capture new developments and applications in emerging fields of computing; and address topics that provide support for computing, such as mathematics, statistics, life and physical sciences, and business.

Published Titles

Paul Anderson, *Web 2.0 and Beyond: Principles and Technologies*

Henrik Bærbak Christensen, *Flexible, Reliable Software: Using Patterns and Agile Development*

John S. Conery, *Explorations in Computing: An Introduction to Computer Science*

Ted Herman, *A Functional Start to Computing with Python*

Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph, *Foundations of Semantic Web Technologies*

Mark J. Johnson, *A Concise Introduction to Data Structures using Java* *Uvais Qidwai and C.H. Chen*, *Digital Image Processing: An Algorithmic Approach with MATLAB®*

Mark J. Johnson, *A Concise Introduction to Programming in Python*

Lisa C. Kaczmarczyk, *Computers and Society: Computing for Good*

Mark C. Lewis, *Introduction to the Art of Programming Using Scala*

Bill Manaris and Andrew R. Brown, *Making Music with Computers: Creative Programming in Python*

Henry M. Walker, *The Tao of Computing, Second Edition*

CHAPMAN & HALL/CRC
TEXTBOOKS IN COMPUTING

Making Music with Computers

Creative Programming in Python

Bill Manaris

College of Charleston
South Carolina, USA

Andrew R. Brown

Queensland University of Technology
Keperra, Australia



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2014 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20140402

International Standard Book Number-13: 978-1-4822-2221-0 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Contents

Foreword, xix

Preface, xxi

The Authors, xxvii

Acknowledgments, xxix

CHAPTER 1 ■ Introduction and History	1
1.1 OVERVIEW	1
1.2 CONNECTING MUSIC, NATURE, AND NUMBER	1
1.2.1 Pythagoras—Harmonic Series	2
1.2.2 The Antikythera Mechanism—The First Known Computer	4
1.2.3 Johannes Kepler—Harmony of the World	4
1.2.4 Cymatics	6
1.2.5 Fractals	7
1.3 COMPUTER MUSIC HISTORY	9
1.3.1 Automated Music	10
1.3.2 Early Computer Music	11
1.3.3 Electronic Music	12
1.3.3.1 <i>Reflection Questions</i>	17
1.4 ALGORITHMS AND PROGRAMMING	17
1.5 THE COMPUTER AS A MUSICAL INSTRUMENT	19
1.6 SOFTWARE USED IN THIS BOOK	21
1.6.1 Case Study: Running a Python Program	22
1.7 SUMMARY	23

CHAPTER 2 ■ Elements of Music and Code	25
2.1 OVERVIEW	25
2.2 MUSIC IS SOUND AND ...	25
2.3 NOTES	26
2.3.1 Musical Notation	27
2.3.2 Pitch	28
2.3.2.1 <i>Pitches Are Integers</i>	28
2.3.3 Duration	29
2.3.3.1 <i>Durations Are Real Numbers</i>	31
2.3.4 Dynamic	31
2.3.5 Panning	31
2.3.6 Creating Notes	32
2.4 RESTS	33
2.4.1 Creating Rests	34
2.4.2 Case Study: Playing a Note	34
2.4.2.1 <i>Comments</i>	35
2.4.3 Exercise	36
2.5 VARIABLES AND ASSIGNMENT	36
2.5.1 Examples	37
2.5.2 Reserved Words	38
2.6 NUMBERS	39
2.6.1 Integers	39
2.6.2 Floats	40
2.6.3 Arithmetic Expressions	40
2.7 INPUT AND OUTPUT	42
2.7.1 Input from the Keyboard	42
2.7.2 Output to the Screen	43
2.8 DATA TYPES	44
2.8.1 The type() Function	44
2.8.2 Case Study: Finding the Octave of a Pitch	45
2.8.3 Testing Programs	46
2.8.4 Exercise	46
2.9 SUMMARY	47

CHAPTER 3 ■ Organization and Data	49
3.1 OVERVIEW	49
3.2 MUSICAL ORGANIZATION	49
3.2.1 Music Data Structure	50
3.3 PHRASES	51
3.3.1 Creating Phrases	51
3.3.2 Adding Notes	52
3.4 PYTHON LISTS	53
3.4.1 List Concatenation	54
3.4.2 List Repetition	54
3.5 ADDING NOTES WITH LISTS	55
3.6 CASE STUDY: LUDWIG VAN BEETHOVEN—“FÜR ELISE”	56
3.6.1 Exercise	57
3.7 MUSICAL SCALES	57
3.7.1 The Major Scale	58
3.7.2 The Minor Scale	59
3.7.3 Other Scales	59
3.7.4 Exercise	60
3.8 MUSICAL INSTRUMENTS	60
3.8.1 MIDI Instruments	61
3.9 SETTING THE INSTRUMENT	61
3.9.1 Exercise	62
3.9.1.1 <i>Setting the Tempo</i>	63
3.10 CASE STUDY: HAROLD FALTERMEYER—“AXEL F”	63
3.10.1 Exercises	64
3.11 CHORDS	64
3.11.1 Adding Chords	65
3.11.2 Case Study: Bruce Hornsby—“The Way It Is”	67
3.11.3 Adding Chords with Lists	68
3.11.4 Case Study: 2Pac—“Changes”	68
3.12 PARTS	69
3.12.1 Creating Parts	70
3.12.2 MIDI Channels	71

3.12.3	Adding Phrases	71
3.12.4	Creating Ensembles	72
3.13	SCORES	74
3.13.1	Creating Scores	74
3.13.2	Putting It All Together	75
3.14	A COMPLETE EXAMPLE	76
3.14.1	Case Study: Joseph Kosma—“Autumn Leaves” (Jazz Trio)	76
3.14.2	Exercise	79
3.15	MIDI DRUMS AND PERCUSSIVE SOUNDS	79
3.15.1	Exercises	80
3.15.2	Case Study: Drum Machines	80
3.15.2.1	<i>Drum Machine Pattern #1</i>	81
3.15.2.2	<i>Exercise</i>	83
3.15.3	Case Study: Deep Purple—“Smoke on the Water”	83
3.16	TOP-DOWN DESIGN	84
3.17	INPUT AND OUTPUT	85
3.17.1	Reading MIDI Files	85
3.17.2	Writing MIDI Files	86
3.17.3	Exercises	87
3.18	SUMMARY	87
CHAPTER 4 ■ Transformation and Process		89
4.1	OVERVIEW	89
4.2	GESTURES, EMOTION, AND MUSICAL STRUCTURE	89
4.2.1	Musical Patterns	90
4.3	MINIMALISM	92
4.3.1	Repetition and Phasing	93
4.3.2	Case Study: Steve Reich, “Piano Phase” (1967)	93
4.4	MODIFYING MUSICAL MATERIAL (MOD FUNCTIONS)	95
4.4.1	Modifying Volume	96
4.4.2	Modifying Duration	96
4.4.3	Modifying Pitch	97
4.4.4	Modifying with Randomness	98

4.5	MUSICAL CANON	98
4.5.1	Case Study: Traditional “Row Your Boat”	99
4.5.1.1	<i>Exercise</i>	101
4.5.2	Analyzing the Musical Process	101
4.5.2.1	<i>Creating Musical Material</i>	102
4.5.2.2	<i>Making Copies of Musical Material</i>	102
4.5.2.3	<i>Shifting Musical Material in Time</i>	103
4.5.2.4	<i>Transposing Musical Material</i>	103
4.5.2.5	<i>Combining Music Material</i>	103
4.5.2.6	<i>Saving and Playing Musical Material</i>	105
4.5.3	Case Study: J.S. Bach—Goldberg Ground, Canon 1 (BWV 1087)	105
4.5.4	Case Study: Trias Harmonica canon (BWV 1072)	107
4.5.5	Exercises	110
4.5.6	Case Study: Arvo Pärt—“Cantus in Memoriam” (1977)	110
4.5.7	Exercises	112
4.6	VIEWING MUSIC	112
4.6.1	Notation Display	113
4.6.2	Piano Roll Display	113
4.6.3	Internal Values Display	114
4.6.4	Sketch Display	116
4.6.5	Exercises	116
4.7	THE SOFTWARE DEVELOPMENT PROCESS	117
4.7.1	Design	117
4.7.2	Implementation	118
4.7.3	Testing	118
4.7.4	Documentation—Good Style and Comments	119
4.8	CASE STUDY: COMPUTER-AIDED MUSIC COMPOSITION	120
4.8.1	Exercise	123
4.9	SUMMARY	123

CHAPTER 5 ■ Iteration and Lists	125
5.1 OVERVIEW	125
5.2 ITERATION	125
5.2.1 The Python for Loop	125
5.2.2 Exercises	127
5.3 CASE STUDY: ARPEGGIATORS	128
5.3.1 Arpeggiator #1—Using Absolute Pitches	128
5.3.2 Constants	129
5.3.2.1 Exercise	131
5.3.3 Interactive Processes	131
5.3.4 Arpeggiator #2—Using Relative Pitches	131
5.3.4.1 Exercises	133
5.4 PYTHON LIST OPERATIONS	133
5.4.1 Accessing List Items	134
5.4.2 Modifying List Items	135
5.4.3 List Functions	136
5.4.4 Case Study: Scale Tutor	136
5.4.5 Case Study: Interactive PianoRoll Generator	137
5.4.5.1 Exercises	139
5.4.6 The range() Function	140
5.4.6.1 Exercises	141
5.4.7 The frange() Function	141
5.4.8 Iterating with Lists	142
5.5 ITERATIVE MUSICAL PROCESSES	143
5.5.1 Case Study: Mod Retrograde	143
5.5.2 Exercises	146
5.5.3 Case Study: Guitar Effect, FX-35 Octoplus	147
5.5.4 Exercises	148
5.6 DNA MUSIC	149
5.6.1 Case Study: Protein Music—Human Thymidylate Synthase A	149
5.6.1.1 Exercises	152
5.7 SUMMARY	153

CHAPTER 6 ■ Randomness and Choices	155
6.1 OVERVIEW	155
6.2 RANDOMNESS AND CREATIVITY	155
6.2.1 Case Study: Mozart—“Musikalisches Würfelspiel”	156
6.2.1.1 <i>Exercise</i>	159
6.3 INDETERMINISM AND SERIALISM	160
6.3.1 Case Study: Pierre Cage—“Structures pour deux Chances”	161
6.3.1.1 <i>Exercises</i>	162
6.4 PYTHON RANDOM FUNCTIONS	163
6.4.1 <i>Exercise</i>	164
6.4.2 <code>randint()</code>	164
6.4.3 <code>choice()</code>	165
6.5 STOCHASTIC MUSIC	165
6.5.1 Case Study: Iannis Xenakis—“Concret PH”	166
6.6 HARNESSING (OR SIEVING) RANDOMNESS	168
6.6.1 Case Study: Wind Chimes	169
6.6.1.1 <i>Exercises</i>	170
6.6.2 Case Study: Pentatonic Melody Generator	170
6.6.3 Weighted Probabilities	171
6.7 SELECTION	172
6.7.1 Python if Statement	173
6.7.1.1 <i>Many Cases</i>	174
6.7.2 Case Study: Flipping a Coin	175
6.7.3 Case Study: Russian Roulette	176
6.7.3.1 <i>Exercise</i>	176
6.7.4 Case Study: Throwing Dice	177
6.7.4.1 <i>Nesting “If” Statements</i>	178
6.7.4.2 <i>Exercise</i>	178
6.7.5 Case Study: Let the Drums Come Alive	179
6.7.5.1 <i>Exercises</i>	181
6.8 PYTHON RELATIONAL OPERATORS	181

6.9	PYTHON BOOLEAN VALUES	184
6.10	PYTHON LOGICAL OPERATORS	185
6.10.1	Case Study: Music from Weighted Probabilities	186
6.10.1.1	<i>Exercises</i>	189
6.11	SUMMARY	190
CHAPTER 7 ■ Sonification and Big Data		191
7.1	OVERVIEW	191
7.2	DATA SONIFICATION	192
7.2.1	The mapValue() Function	192
7.2.2	The mapScale() Function	194
7.3	CASE STUDY: KEPLER—“HARMONIES OF THE WORLD” (1619)	196
7.3.1	Exercise	199
7.4	PYTHON STRINGS	200
7.4.1	Case Study: Music from Text	202
7.4.1.1	<i>Exercise</i>	204
7.4.2	String Library Functions	204
7.4.3	Case Study: Guido d’Arezzo—“Word Music” (ca. 1000)	206
7.4.4	Python Nested Loops	208
7.4.5	Exercise	210
7.5	FILE INPUT AND OUTPUT	211
7.5.1	Reading Files	211
7.5.2	Writing Files	212
7.5.3	Exercises	213
7.6	PYTHON WHILE LOOP	213
7.6.1	Exercise	214
7.7	BIG DATA	215
7.7.1	Case Study: Biosignal Sonification	215
7.7.1.1	<i>Sonification Design</i>	217
7.7.1.2	<i>Python Parallel Assignment</i>	220
7.7.2	Exercises	223

7.8	PYTHON FUNCTIONS	223
7.8.1	Defining Functions	224
7.8.2	Exercise	225
7.8.3	Returning Values	226
7.8.4	Exercises	227
7.8.5	Scope of Variables	229
7.9	IMAGE SONIFICATION	229
7.9.1	Python Images	229
7.9.2	Image Library Functions	230
7.9.3	Case Study: Visual Soundscape	231
	7.9.3.1 <i>Sonification Design</i>	232
	7.9.3.2 <i>Defining a Function</i>	236
7.9.4	Python Nested Loops (again)	238
7.9.5	Exercises	239
7.10	SUMMARY	239
CHAPTER 8 ■ Interactive Musical Instruments		241
8.1	OVERVIEW	241
8.2	BUILDING MUSICAL INSTRUMENTS	241
8.3	GRAPHICAL USER INTERFACES	242
8.3.1	Creating Displays	243
8.3.2	Graphics Objects	244
	8.3.2.1 <i>Exercise</i>	245
8.3.3	Showing Display Coordinates	245
8.4	CASE STUDY: RANDOM CIRCLES	246
8.4.1	Exercises	247
8.5	GUI WIDGETS	248
8.5.1	Event-Driven Programming	248
8.5.2	Callback Functions	249
8.6	CASE STUDY: A SIMPLE MUSICAL INSTRUMENT	250
8.6.1	Python Global Statement	252
8.6.2	Exercise	253
8.7	PLAY CLASS	253

8.8	CASE STUDY: AN AUDIO INSTRUMENT FOR CONTINUOUS PITCH CONTROL	255
8.9	AUDIOSAMPLE CLASS	258
8.9.1	Creating Audio Samples	258
8.9.2	Exercise	260
8.10	MIDISEQUENCE CLASS	260
8.10.1	Creating MIDI Sequences	261
8.10.2	Exercises	262
8.11	PAPER PROTOTYPING	262
8.12	A SIMPLE METHODOLOGY FOR DEVELOPING GUIs	263
8.12.1	Listen, Listen, Listen	265
8.13	EVENT HANDLING	265
8.13.1	Keyboard Events	266
8.13.2	Mouse Events	266
8.13.2.1	<i>Example</i>	266
8.13.2.2	<i>Exercises</i>	268
8.13.3	Case Study: Drawing Musical Circles	268
8.13.3.1	<i>Defining Callback Functions</i>	271
8.13.3.2	<i>Exercises</i>	273
8.14	CASE STUDY: A VIRTUAL PIANO	273
8.14.1	Exercise	278
8.14.2	A Variation, Using Parallel Lists	278
8.14.2.1	<i>Exercises</i>	281
8.15	SCHEDULING FUTURE EVENTS	282
8.15.1	Case Study: Random Circles with Timer	282
8.15.2	The Timer Class	286
8.15.2.1	<i>Creating Timers</i>	286
8.16	SUMMARY	287
CHAPTER 9 ■ Making Connections		289
9.1	OVERVIEW	289
9.2	MIDI DEVICES—CONNECTING TO PIANOS, GUITARS, ETC.	290

9.2.1	Case Study: Make Music with a MIDI Instrument	291
9.2.1.1	<i>Exercise</i>	294
9.2.2	The MIDI Library	294
9.2.2.1	<i>The MidiIn Class</i>	294
9.2.2.2	<i>The MidiOut Class</i>	296
9.3	OSC DEVICES—CONNECTING TO SMARTPHONES, TABLETS, ETC.	298
9.3.1	OSC Messages	299
9.3.2	Case Study: Hello (OSC) World!	299
9.3.2.1	<i>Program for OSC Server Device</i>	299
9.3.2.2	<i>Program for OSC Client Device</i>	301
9.3.2.3	<i>Exercises</i>	301
9.3.3	The OSC Library	302
9.3.3.1	<i>The OscIn Class</i>	302
9.3.3.2	<i>The OscOut Class</i>	304
9.3.4	Case Study: Make Music with your Smartphone	305
9.3.4.1	<i>Performance Instructions</i>	305
9.3.4.2	<i>Setting up Your Smartphone (OSC Client)</i>	307
9.3.4.3	<i>Setting up Your Computer (OSC Server)</i>	307
9.3.4.4	<i>Exercises</i>	312
9.3.5	Hybrid Musical Instrument Projects	313
9.4	SUMMARY	313
CHAPTER 10 ■ Music, Number, and Nature		315
10.1	OVERVIEW	315
10.2	ORIGINS AND REPRESENTATIONS	316
10.2.1	Pythagorean Theorem	317
10.2.2	Python as a Representation	318
10.3	CASE STUDY: MUSIC FROM MATH CURVES	319
10.3.1	Hearing the Music	320
10.3.2	Exercises	323
10.4	MATH LIBRARY	324
10.5	CASE STUDY: THE HARMONOGRAPH	324

10.5.1	Lateral Harmonograph	327
10.5.2	Rotary Harmonograph	330
10.5.3	Exercises	333
10.5.4	Noninteger Ratios	333
10.6	CASE STUDY: KEPLER'S HARMONY OF THE WORLD, NO. 2	334
10.6.1	Exercises	337
10.7	SUMMARY	338
CHAPTER 11 ■ Exploring Powerful Ideas		339
<hr/>		
11.1	OVERVIEW	339
11.2	FRACTALS AND RECURSION	340
11.3	FIBONACCI NUMBERS AND THE GOLDEN RATIO	340
11.3.1	Case Study: The Golden Tree	343
11.3.1.1	<i>Exercises</i>	347
11.4	ZIPF'S LAW	348
11.4.1	Zipf's Law and Music	350
11.4.2	What Does It Mean?	351
11.4.3	Measuring Zipf Proportions	352
11.4.3.1	<i>Top-Down Design (Revisited)</i>	354
11.4.4	Python Dictionaries	356
11.4.5	Exercises	358
11.5	PYTHON CLASSES	358
11.6	CASE STUDY: THE NOTE CLASS	359
11.6.1	Creating Note Objects	361
11.6.2	Defining the Class	361
11.6.2.1	<i>Checking for Data Integrity</i>	364
11.6.3	Python Exceptions	366
11.6.4	Exercises	366
11.7	CASE STUDY: A SLIDER CONTROL	367
11.7.1	Creating SliderControl Objects	368
11.7.2	Defining the Class	369
11.7.3	Exercises	370

11.8	ANIMATION	371
11.8.1	Frame Rate	372
11.8.2	Case Study: A Revolving Musical Sphere	372
11.8.2.1	<i>Color Gradients</i>	373
11.8.3	Defining the Class	374
11.8.3.1	<i>Spherical Coordinate System</i>	375
11.8.4	Exercises	383
11.9	CYMATICS	384
11.9.1	Vectors and Python Complex Numbers	387
11.9.2	Defining the Boid Universe	389
11.9.3	Defining the Boids	393
11.9.3.1	<i>Boid Sensing</i>	396
11.9.3.2	<i>Boid Acting</i>	398
11.9.4	Creating the Simulation	398
11.10	EXERCISES	399
11.11	SUMMARY	402

REFERENCES, 405

APPENDIX A: MIDI CONSTANTS, 409

APPENDIX B: MUSIC LIBRARY FUNCTIONS, 419

APPENDIX C: GUI LIBRARY FUNCTIONS, 429

APPENDIX D: OTHER FUNCTIONS, 449

Foreword

THE HUMAN DESIRE TO EXPRESS and communicate has influenced computing almost as long as there have been computers. ENIAC was first turned on in 1947. The first computer music was generated in 1957.

The desire to *say* more with a computer has driven many advances in computer science. Ivan Sutherland invented interactive computer graphics in 1963, and his creation inspired the idea of classes in object-oriented programming. Alan Kay and Adele Goldberg described the computer as human's first *meta-medium*, the first creative medium that could encompass all previous media. Their research group at Xerox's Palo Alto Research Center (PARC) worked in the 1970s to answer the question, "What would a computer used for creative expression look like?" That's what led them to invent the desktop user interface as we know it today. In a real sense, the menus and windows that we use today to access Facebook were invented in order to make the most powerful tool ever for human expression.

Making music on a computer is a natural way to learn more about mathematics, computer science, and music. Bill Manaris and Andrew Brown have created this marvelous book that will engage and inspire you to learn more about the science and art of creating music through computation. They lead us through exploration of fascinating questions. How does music draw on both mathematical patterns and randomness? How did Bach use algorithms to generate canons? How can we turn data about proteins and planets into music? What kinds of new interfaces can you create to make it easier for you and others to make music?

Bill and Andrew offer an accessible path into a wonderful world that is both as modern as your new laptop and as ancient as Plato. In that world of music and mathematics, they constructed a sandbox of computational

tools. They encourage you to create, to compose music, and to play with patterns and data. They invite you to continue in the traditions of Ivan Sutherland and Alan Kay to use computing to explore powerful and creative ideas.

Mark Guzdial

Georgia Institute of Technology

July 2013

Preface

THE BOOK IN YOUR HANDS is the result of more than a decade of independent and collaborative effort by the two authors and their computer music associates. Combining computers and music has a long and fruitful heritage. The ideas which underpin the connection between calculating and composing date back centuries. In the 21st century, computers and music are more closely aligned than ever before. In particular, computers have become indispensable in music making, distribution, performance, and consumption.

This book introduces important concepts and skills necessary to make music with computers. It interweaves computing pedagogy with musical concepts and creative activities. It does this while maintaining a natural, steady increase in computational skills that are motivated by creative musical contexts.

This book is intended primarily for introductory computer science courses and for courses in the intersection of computing and the arts. However, it is naturally suited for self-study. It assumes little musical and programming experience; it introduces topics and concepts as they arise through motivating, and hopefully inspiring examples.

CREATIVE PROGRAMMING

“Making Music with Computers” is an introduction to creative software development in the Python programming language. It uses music-making as a vehicle to introduce computer programming and computational thinking to non-traditional audiences. This book helps computer science educators teach students how to synthesize the creativity and design of the arts with the mathematical rigor and formality of computer science.

Initially inspired by Randy Pausch’s “head-fake” approach^{*}, we utilize exciting and innovative music-creation activities to ultimately teach

^{*} See Randy Pausch’s “Last Lecture” (readily available online).

introductory computer science concepts. Our goal is to keep this “game” going throughout the book, just long enough so that the students learn to express themselves algorithmically.

The book covers all concepts found in a traditional “Intro to Computer Programming” (CS1) course. These concepts include data types, variables, assignment, arithmetic operators, input/output, algorithms, selection (`if` statements), relational operators, logical operators, iteration (loops), lists (arrays), functions, modularization (functions), classes (object-oriented programming). Additionally, the book covers graphical user interfaces (GUIs), event-driven programming, big data, MIDI programming, client-server programming (via OSC messages), recursion, fractals, and complex system dynamics (boids).

TARGET AUDIENCE

This book addresses two trends in computing education: (1) the growing use of the Python language for teaching introductory programming, and (2) the increasing infusion of computational thinking into liberal arts courses, especially interdisciplinary offerings in computing and the arts. It does so by presenting computer music topics in an accessible way for our two main target audiences:

- First- and second-year university students, as well as advanced high school students, who are interested in computer music and wish to learn computer programming in a creative context; and
- Musicians of all levels and backgrounds who wish to expand their creative horizons by modeling musical processes through computer programming, and by applying these processes to create novel and intriguing musical material for composition and live performance.

NAVIGATING THE BOOK

The book may be navigated using one of three narratives, *objects first*, *procedures first*, or *à la carte*:

- **Objects first** (chapters 1–3, followed by chapters 8–11, with just-in-time introduction of `for` loops, functions, and `if` statements). This approach works well with inexperienced students, as it is creatively rich. It includes building graphical user interfaces (GUIs) and interactive musical instruments, and thus motivates hard-to-grasp

- [read Scandal's Daughter \(Series, Book 1\) pdf](#)
- [download online Catalyst \(Barque Cats, Book 3\)](#)
- [read Cruel Rider](#)
- [read online The Muse Is Always Half-Dressed in New Orleans: and Other Essays](#)
- [DANGER OF SELF INTERPRETATION OF QURAN pdf, azw \(kindle\), epub](#)
- [download online You Can't Make This Up: Miracles, Memories, and the Perfect Marriage of Sports and Television](#)

- <http://xn--d1aboelcb1f.xn--p1ai/lib/Writing-Secure-Code--2nd-Edition-.pdf>
- <http://korplast.gr/lib/Women-on-the-Verge-of-a-Nervous-Breakdown--BFI-Modern-Classics-.pdf>
- <http://econtact.webschaefer.com/?books/Cruel-Rider.pdf>
- <http://www.1973vision.com/?library/Acting-in-an-Uncertain-World--An-Essay-on-Technical-Democracy--Inside-Technology-.pdf>
- <http://patrickvincitore.com/?ebooks/CCENT-CCNA-ICND1-Official-Exam-Certification-Guide.pdf>
- <http://betsy.wesleychapelcomputerrepair.com/library/Postmodern-Approaches-to-the-Short-Story--Contributions-to-the-Study-of-World-Literature--Book-118-.pdf>