

Contents

1	Computing with Formulas	1
1.1	The First Programming Encounter: A Formula	1
1.1.1	Using a Program as a Calculator	2
1.1.2	About Programs and Programming	2
1.1.3	Tools for Writing Programs	3
1.1.4	Using Idle to Write the Program	4
1.1.5	How to Run the Program	7
1.1.6	Verifying the Result	8
1.1.7	Using Variables	8
1.1.8	Names of Variables	9
1.1.9	Reserved Words in Python	10
1.1.10	Comments	10
1.1.11	Formatting Text and Numbers	11
1.2	Computer Science Glossary	14
1.3	Another Formula: Celsius-Fahrenheit Conversion	19
1.3.1	Potential Error: Integer Division	19
1.3.2	Objects in Python	20
1.3.3	Avoiding Integer Division	21
1.3.4	Arithmetic Operators and Precedence	22
1.4	Evaluating Standard Mathematical Functions	22
1.4.1	Example: Using the Square Root Function	23
1.4.2	Example: Using More Mathematical Functions	25
1.4.3	A First Glimpse of Round-Off Errors	25
1.5	Interactive Computing	26
1.5.1	Calculating with Formulas in the Interactive Shell	27
1.5.2	Type Conversion	28
1.5.3	IPython	29
1.6	Complex Numbers	31

1.6.1	Complex Arithmetics in Python	32
1.6.2	Complex Functions in Python	33
1.6.3	Unified Treatment of Complex and Real Functions	33
1.7	Summary	35
1.7.1	Chapter Topics	35
1.7.2	Summarizing Example: Trajectory of a Ball	38
1.7.3	About Typesetting Conventions in This Book ..	40
1.8	Exercises	41
2	Loops and Lists	51
2.1	While Loops	51
2.1.1	A Naive Solution	51
2.1.2	While Loops	52
2.1.3	Boolean Expressions	54
2.1.4	Loop Implementation of a Sum	56
2.2	Lists	57
2.2.1	Basic List Operations	58
2.2.2	For Loops	60
2.3	Alternative Implementations with Lists and Loops	62
2.3.1	While Loop Implementation of a For Loop	62
2.3.2	The Range Construction	62
2.3.3	For Loops with List Indices	63
2.3.4	Changing List Elements	65
2.3.5	List Comprehension	65
2.3.6	Traversing Multiple Lists Simultaneously	66
2.4	Nested Lists	66
2.4.1	A Table as a List of Rows or Columns	67
2.4.2	Printing Objects	68
2.4.3	Extracting Sublists	69
2.4.4	Traversing Nested Lists	71
2.5	Tuples	73
2.6	Summary	74
2.6.1	Chapter Topics	74
2.6.2	Summarizing Example: Analyzing List Data ...	77
2.6.3	How to Find More Python Information	80
2.7	Exercises	81
3	Functions and Branching	91
3.1	Functions	91
3.1.1	Functions of One Variable	91
3.1.2	Local and Global Variables	93
3.1.3	Multiple Arguments	95
3.1.4	Multiple Return Values	97
3.1.5	Functions with No Return Values	99

3.1.6	Keyword Arguments	100
3.1.7	Doc Strings	103
3.1.8	Function Input and Output	104
3.1.9	Functions as Arguments to Functions	104
3.1.10	The Main Program	106
3.1.11	Lambda Functions	107
3.2	Branching	108
3.2.1	If-Else Blocks	108
3.2.2	Inline If Tests	110
3.3	Summary	111
3.3.1	Chapter Topics	111
3.3.2	Summarizing Example: Numerical Integration ..	113
3.4	Exercises	116
4	Input Data and Error Handling	129
4.1	Asking Questions and Reading Answers	130
4.1.1	Reading Keyboard Input	130
4.1.2	The Magic “eval” Function	131
4.1.3	The Magic “exec” Function	135
4.1.4	Turning String Expressions into Functions	136
4.2	Reading from the Command Line	137
4.2.1	Providing Input on the Command Line	137
4.2.2	A Variable Number of Command-Line Arguments	138
4.2.3	More on Command-Line Arguments	139
4.2.4	Option–Value Pairs on the Command Line	140
4.3	Handling Errors	142
4.3.1	Exception Handling	143
4.3.2	Raising Exceptions	146
4.4	A Glimpse of Graphical User Interfaces	148
4.5	Making Modules	151
4.5.1	Example: Compound Interest Formulas	152
4.5.2	Collecting Functions in a Module File	153
4.5.3	Using Modules	157
4.6	Summary	159
4.6.1	Chapter Topics	159
4.6.2	Summarizing Example: Bisection Root Finding .	162
4.7	Exercises	170
5	Array Computing and Curve Plotting	177
5.1	Vectors	178
5.1.1	The Vector Concept	178
5.1.2	Mathematical Operations on Vectors	179
5.1.3	Vector Arithmetics and Vector Functions	181
5.2	Arrays in Python Programs	183

5.2.1	Using Lists for Collecting Function Data	183
5.2.2	Basics of Numerical Python Arrays	184
5.2.3	Computing Coordinates and Function Values . . .	185
5.2.4	Vectorization	186
5.3	Curve Plotting	188
5.3.1	Matplotlib; Pylab	188
5.3.2	Matplotlib; Pyplot	192
5.3.3	SciTools and Easyviz	194
5.3.4	Making Animations	199
5.3.5	Curves in Pure Text	203
5.4	Plotting Difficulties	204
5.4.1	Piecewisely Defined Functions	205
5.4.2	Rapidly Varying Functions	210
5.4.3	Vectorizing StringFunction Objects	211
5.5	More on Numerical Python Arrays	212
5.5.1	Copying Arrays	212
5.5.2	In-Place Arithmetics	213
5.5.3	Allocating Arrays	214
5.5.4	Generalized Indexing	214
5.5.5	Testing for the Array Type	215
5.5.6	Compact Syntax for Array Generation	216
5.5.7	Shape Manipulation	216
5.6	Higher-Dimensional Arrays	217
5.6.1	Matrices and Arrays	217
5.6.2	Two-Dimensional Numerical Python Arrays . . .	218
5.6.3	Array Computing	220
5.6.4	Two-Dimensional Arrays and Functions of Two Variables	221
5.6.5	Matrix Objects	222
5.7	Summary	223
5.7.1	Chapter Topics	223
5.7.2	Summarizing Example: Animating a Function . .	224
5.8	Exercises	229
6	Files, Strings, and Dictionaries	243
6.1	Reading Data from File	243
6.1.1	Reading a File Line by Line	244
6.1.2	Reading a Mixture of Text and Numbers	247
6.1.3	What Is a File, Really?	248
6.2	Dictionaries	252
6.2.1	Making Dictionaries	252
6.2.2	Dictionary Operations	253
6.2.3	Example: Polynomials as Dictionaries	254
6.2.4	Example: File Data in Dictionaries	256
6.2.5	Example: File Data in Nested Dictionaries . . .	257

6.2.6	Example: Comparing Stock Prices	262
6.3	Strings	266
6.3.1	Common Operations on Strings	266
6.3.2	Example: Reading Pairs of Numbers	270
6.3.3	Example: Reading Coordinates	272
6.4	Reading Data from Web Pages	274
6.4.1	About Web Pages	274
6.4.2	How to Access Web Pages in Programs	276
6.4.3	Example: Reading Pure Text Files	277
6.4.4	Example: Extracting Data from an HTML Page	278
6.5	Writing Data to File	279
6.5.1	Example: Writing a Table to File	280
6.5.2	Standard Input and Output as File Objects	281
6.5.3	Reading and Writing Spreadsheet Files	284
6.6	Summary	289
6.6.1	Chapter Topics	289
6.6.2	Summarizing Example: A File Database	291
6.7	Exercises	294
7	Introduction to Classes	305
7.1	Simple Function Classes	306
7.1.1	Problem: Functions with Parameters	306
7.1.2	Representing a Function as a Class	308
7.1.3	Another Function Class Example	314
7.1.4	Alternative Function Class Implementations	315
7.1.5	Making Classes Without the Class Construct ...	317
7.2	More Examples on Classes	320
7.2.1	Bank Accounts	320
7.2.2	Phone Book	322
7.2.3	A Circle	323
7.3	Special Methods	324
7.3.1	The Call Special Method	325
7.3.2	Example: Automagic Differentiation	325
7.3.3	Example: Automagic Integration	328
7.3.4	Turning an Instance into a String	330
7.3.5	Example: Phone Book with Special Methods ...	331
7.3.6	Adding Objects	333
7.3.7	Example: Class for Polynomials	333
7.3.8	Arithmetic Operations and Other Special Methods	337
7.3.9	Special Methods for String Conversion	338
7.4	Example: Class for Vectors in the Plane	340
7.4.1	Some Mathematical Operations on Vectors	340
7.4.2	Implementation	340
7.4.3	Usage	342

7.5	Example: Class for Complex Numbers	343
7.5.1	Implementation	344
7.5.2	Illegal Operations	345
7.5.3	Mixing Complex and Real Numbers	346
7.5.4	Special Methods for “Right” Operands	348
7.5.5	Inspecting Instances	350
7.6	Static Methods and Attributes	351
7.7	Summary	352
7.7.1	Chapter Topics	352
7.7.2	Summarizing Example: Interval Arithmetics	353
7.8	Exercises	359
8	Random Numbers and Simple Games	375
8.1	Drawing Random Numbers	376
8.1.1	The Seed	376
8.1.2	Uniformly Distributed Random Numbers	377
8.1.3	Visualizing the Distribution	378
8.1.4	Vectorized Drawing of Random Numbers	379
8.1.5	Computing the Mean and Standard Deviation ..	380
8.1.6	The Gaussian or Normal Distribution	381
8.2	Drawing Integers	382
8.2.1	Random Integer Functions	383
8.2.2	Example: Throwing a Die	384
8.2.3	Drawing a Random Element from a List	385
8.2.4	Example: Drawing Cards from a Deck	385
8.2.5	Example: Class Implementation of a Deck	387
8.3	Computing Probabilities	390
8.3.1	Principles of Monte Carlo Simulation	390
8.3.2	Example: Throwing Dice	391
8.3.3	Example: Drawing Balls from a Hat	393
8.3.4	Example: Policies for Limiting Population Growth	395
8.4	Simple Games	398
8.4.1	Guessing a Number	398
8.4.2	Rolling Two Dice	398
8.5	Monte Carlo Integration	401
8.5.1	Standard Monte Carlo Integration	401
8.5.2	Area Computing by Throwing Random Points ..	404
8.6	Random Walk in One Space Dimension	406
8.6.1	Basic Implementation	406
8.6.2	Visualization	407
8.6.3	Random Walk as a Difference Equation	408
8.6.4	Computing Statistics of the Particle Positions ..	408
8.6.5	Vectorized Implementation	409
8.7	Random Walk in Two Space Dimensions	411

8.7.1	Basic Implementation	411
8.7.2	Vectorized Implementation	413
8.8	Summary	414
8.8.1	Chapter Topics	414
8.8.2	Summarizing Example: Random Growth.....	415
8.9	Exercises	421
9	Object-Oriented Programming	437
9.1	Inheritance and Class Hierarchies	437
9.1.1	A Class for Straight Lines.....	438
9.1.2	A First Try on a Class for Parabolas	439
9.1.3	A Class for Parabolas Using Inheritance	439
9.1.4	Checking the Class Type	441
9.1.5	Attribute versus Inheritance.....	442
9.1.6	Extending versus Restricting Functionality	443
9.1.7	Superclass for Defining an Interface	444
9.2	Class Hierarchy for Numerical Differentiation	446
9.2.1	Classes for Differentiation	447
9.2.2	A Flexible Main Program	450
9.2.3	Extensions	451
9.2.4	Alternative Implementation via Functions.....	454
9.2.5	Alternative Implementation via Functional Programming	455
9.2.6	Alternative Implementation via a Single Class ..	456
9.3	Class Hierarchy for Numerical Integration	458
9.3.1	Numerical Integration Methods	458
9.3.2	Classes for Integration.....	459
9.3.3	Using the Class Hierarchy.....	463
9.3.4	About Object-Oriented Programming	466
9.4	Class Hierarchy for Geometric Shapes	467
9.4.1	Using the Class Hierarchy.....	467
9.4.2	Overall Design of the Class Hierarchy	469
9.4.3	The Drawing Tool	470
9.4.4	Implementation of Shape Classes	472
9.4.5	Scaling, Translating, and Rotating a Figure	476
9.5	Summary	480
9.5.1	Chapter Topics	480
9.5.2	Summarizing Example: Input Data Reader.....	482
9.6	Exercises	488
A	Sequences and Difference Equations	497
A.1	Mathematical Models Based on Difference Equations ..	498
A.1.1	Interest Rates	499
A.1.2	The Factorial as a Difference Equation	501
A.1.3	Fibonacci Numbers	502

A.1.4	Growth of a Population	503
A.1.5	Logistic Growth	504
A.1.6	Payback of a Loan	506
A.1.7	Taylor Series as a Difference Equation	507
A.1.8	Making a Living from a Fortune	508
A.1.9	Newton's Method	509
A.1.10	The Inverse of a Function	513
A.2	Programming with Sound	515
A.2.1	Writing Sound to File	515
A.2.2	Reading Sound from File	516
A.2.3	Playing Many Notes	517
A.2.4	Music of a Sequence	518
A.3	Exercises	521
B	Introduction to Discrete Calculus	529
B.1	Discrete Functions	529
B.1.1	The Sine Function	530
B.1.2	Interpolation	532
B.1.3	Evaluating the Approximation	532
B.1.4	Generalization	533
B.2	Differentiation Becomes Finite Differences	535
B.2.1	Differentiating the Sine Function	536
B.2.2	Differences on a Mesh	536
B.2.3	Generalization	538
B.3	Integration Becomes Summation	539
B.3.1	Dividing into Subintervals	540
B.3.2	Integration on Subintervals	541
B.3.3	Adding the Subintervals	542
B.3.4	Generalization	543
B.4	Taylor Series	545
B.4.1	Approximating Functions Close to One Point... ..	545
B.4.2	Approximating the Exponential Function	545
B.4.3	More Accurate Expansions	546
B.4.4	Accuracy of the Approximation	548
B.4.5	Derivatives Revisited	550
B.4.6	More Accurate Difference Approximations	551
B.4.7	Second-Order Derivatives	553
B.5	Exercises	555
C	Introduction to Differential Equations	561
C.1	The Simplest Case	562
C.2	Exponential Growth	564
C.3	Logistic Growth	569
C.4	A Simple Pendulum	570
C.5	A Model for the Spread of a Disease	573

C.6 Exercises	575
D A Complete Differential Equation Project	577
D.1 About the Problem: Motion and Forces in Physics	577
D.1.1 The Physical Problem	577
D.1.2 The Computational Algorithm	580
D.1.3 Derivation of the Mathematical Model	580
D.1.4 Derivation of the Algorithm	583
D.2 Program Development and Testing	584
D.2.1 Implementation	584
D.2.2 Callback Functionality	587
D.2.3 Making a Module	588
D.2.4 Verification	589
D.3 Visualization	591
D.3.1 Simultaneous Computation and Plotting	591
D.3.2 Some Applications	594
D.3.3 Remark on Choosing Δt	594
D.3.4 Comparing Several Quantities in Subplots	595
D.3.5 Comparing Approximate and Exact Solutions ..	596
D.3.6 Evolution of the Error as Δt Decreases	597
D.4 Exercises	601
E Programming of Differential Equations	603
E.1 Scalar Ordinary Differential Equations	604
E.1.1 Examples on Right-Hand-Side Functions	604
E.1.2 The Forward Euler Scheme	606
E.1.3 Function Implementation	607
E.1.4 Verifying the Implementation	607
E.1.5 Switching Numerical Method	608
E.1.6 Class Implementation	609
E.1.7 Example: Logistic Growth	612
E.2 Systems of Ordinary Differential Equations	613
E.2.1 Mathematical Problem	614
E.2.2 Example of a System of ODEs	615
E.2.3 From Scalar ODE Code to Systems	616
E.2.4 Numerical Methods	619
E.2.5 The ODE Solver Class Hierarchy	621
E.2.6 The Backward Euler Method	623
E.2.7 Application 1: $u' = u$	626
E.2.8 Application 2: The Logistic Equation	627
E.2.9 Application 3: An Oscillating System	629
E.2.10 Application 4: The Trajectory of a Ball	631
E.3 Exercises	633
F Debugging	655
F.1 Using a Debugger	655

F.2	How to Debug	658
F.2.1	A Recipe for Program Writing and Debugging ..	658
F.2.2	Application of the Recipe	660
G	Technical Topics	673
G.1	Different Ways of Running Python Programs	673
G.1.1	Executing Python Programs in IPython	673
G.1.2	Executing Python Programs on Unix	673
G.1.3	Executing Python Programs on Windows	675
G.1.4	Executing Python Programs on Macintosh	677
G.1.5	Making a Complete Stand-Alone Executable ...	677
G.2	Integer and Float Division	677
G.3	Visualizing a Program with Lumpy	678
G.4	Doing Operating System Tasks in Python	681
G.5	Variable Number of Function Arguments	683
G.5.1	Variable Number of Positional Arguments	683
G.5.2	Variable Number of Keyword Arguments	686
G.6	Evaluating Program Efficiency	688
G.6.1	Making Time Measurements	688
G.6.2	Profiling Python Programs	690
	Bibliography	693
	Index	695