About the Author

Kenneth A. Lambert is a professor of computer science and the chair of that department at Washington and Lee University. He has taught introductory programming courses for 29 years and has been an active researcher in computer science education. Lambert has authored or coauthored 25 textbooks, including a series of introductory C++ textbooks with Douglas Nance and Thomas Naps, a series of introductory Java textbooks with Martin Osborne, and a series of introductory Python textbooks. His most recent textbook is Fundamentals of Python: Data Structures.
Contents

Introduction ................................................................. xiii

Chapter 1  Getting Started with Python  ........................................ 1
Taking Care of Preliminaries .................................................. 1
  Downloading and Installing Python ...................................... 1
  Launching and Working in the IDLE Shell ................................. 2
  Obtaining Python Help .................................................. 3
Working with Numbers .................................................... 4
  Using Arithmetic ................................................................ 4
  Working with Variables and Assignment ................................ 6
  Using Functions ................................................................ 9
  Using the math Module .................................................. 9
  Detecting Errors .......................................................... 11
Working with Strings ....................................................... 12
  String Literals ............................................................ 12
  The len, str, int, and float Functions ................................ 14
  Input and Output Functions ............................................... 15
  Indexing, Slicing, and Concatenation ................................... 16
  String Methods .......................................................... 18
Working with Lists ............................................................. 19
  List Literals and Operators ............................................. 19
  List Methods ............................................................ 20
  Lists from Other Sequences ............................................. 22
  Lists and the random Module .......................................... 23
  Tuples as Immutable Lists ............................................. 24
## Making Choices: Selection Statements

- The One-Way if Statement ........................................... 63
- The Two-Way if Statement ........................................... 63
- Probable Options with `random.randint` ......................... 66
- The Multiway if Statement ........................................... 67

### Using Selection to Control Iteration

- The while Loop .......................................................... 69

## Random Walks in Turtle Graphics

- The main Function ...................................................... 79

## Summary

- Exercises ................................................................... 74

## Exercises

- Composing, Saving, and Running Programs ...................... 75
- Computer Hardware ..................................................... 86
- Computer Software ..................................................... 88

## Summary

- Exercises ................................................................... 91

## Defining Functions ..................................................... 93

- The return Statement .................................................... 96
- Optional, Default, and Keyword Arguments ....................... 98
## Functions as General Solutions to Problems
- Regular Polygons
- Functions as Arguments
- Building Functions with lambda Expressions

## Modules as Libraries of Functions
- Math Topic: Graphing Functions
- Functions in Mathematics
- Graphing Functions in Turtle Graphics

## Refactoring a Program with Functions
- Simplifying the Code for the Random Walk
- The atEdge Function
- The randomForward Function
- The randomTurn Function
- Another Version of the Random Walk

## Summary

## Exercises

## User Interaction with the Mouse and the Keyboard

### Using Dialog-Based Input
- Input Dialogs in Turtle Graphics
- Input Dialogs for Text
- Input Dialogs for Numbers

### Responding to Mouse Events
- Drawing Line Segments with Mouse Clicks
- How Event Handling Works
- Freehand Drawing by Dragging the Mouse

### Responding to Keyboard Events
- The onkey Function
- A Complete Retro Drawing Program

### Using Module Variables
- Initializing and Using Module Variables
- Tracking the History of Turtle Positions

### Using Two Mouse Buttons
- Adding an Event-Handling Function for the Right Button
- Example 1: Simple Drawing with Random Colors
- Example 2: Drawing and Moving
- Example 3: Drawing, Moving, and Random Colors
- Example 4: Dialogs for Shape Properties

## Summary

## Exercises
Chapter 7  Recursion ............................................. 135
Recursive Design .................................................. 135
Top-Down Design .................................................. 135
Recursive Function Design ...................................... 137
Recursive Function Call Tracing ............................... 139
Recursive Functions and Loops ................................. 140
Infinite Recursion .................................................. 142
Sequential Search of a List ...................................... 143
Binary Search of a List ............................................ 145
Recursive Patterns in Art: Abstract Painting .............. 147
  Design of the Program .......................................... 148
  Performance Tracking ......................................... 148
  The drawRectangle Function ................................. 149
  The mondrian Function ........................................ 149
  The main Function ............................................. 151
  The tracer and update Functions ............................. 152
Recursive Patterns in Nature: Fractals ....................... 153
  What the Program Does ........................................ 155
  Design of the Program .......................................... 155
  Code for the Program .......................................... 156
Summary ............................................................. 157
Exercises ............................................................ 158

Chapter 8  Objects and Classes ................................ 159
Objects, Methods, and Classes in Turtle Graphics ......... 160
  The Turtle Class and Its Methods ............................ 160
  A Random Walk with Several Turtles ......................... 161
A New Class: RegularPolygon .................................. 162
  Design: Determine the Attributes ............................ 163
  Design: Determine the Behavior ................................ 164
  Implementation: The Structure of a Class Definition ...... 166
  Implementation: The __init__ Method ......................... 168
  Implementation: Showing and Hiding ........................ 169
  Implementation: Getting and Setting ........................ 170
  Implementation: Translation, Scaling, and Rotation .......... 171
  Inheritance: Squares and Hexagons as Subclasses ........... 171
New Class: Menu Item ............................................. 172
  Design: Determine the Attributes and Behavior ............. 173
  Implementation and Testing .................................... 173
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Animations .................................................. 189</td>
</tr>
<tr>
<td></td>
<td>Animating the Turtle with a Timer ......................... 189</td>
</tr>
<tr>
<td></td>
<td>Using the ontimer Function .................................. 190</td>
</tr>
<tr>
<td></td>
<td>Scheduling Actions at Regular Intervals ................... 190</td>
</tr>
<tr>
<td></td>
<td>Animating Many Turtles ..................................... 193</td>
</tr>
<tr>
<td></td>
<td>What Is an Animated Turtle? ................................ 193</td>
</tr>
<tr>
<td></td>
<td>Using an Animated Turtle in a Shell Session ............... 195</td>
</tr>
<tr>
<td></td>
<td>Defining the AnimatedTurtle Class ......................... 195</td>
</tr>
<tr>
<td></td>
<td>Sleepy and Speedy as Animated Turtles ..................... 197</td>
</tr>
<tr>
<td></td>
<td>Creating Custom Turtle Shapes ............................. 199</td>
</tr>
<tr>
<td></td>
<td>Creating Simple Shapes ................................... 199</td>
</tr>
<tr>
<td></td>
<td>Creating Compound Shapes ................................. 202</td>
</tr>
<tr>
<td></td>
<td>Summary ................. .................................... 203</td>
</tr>
<tr>
<td></td>
<td>Exercises .................. ................................ 204</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Turtle Graphics Commands .............................. 205</td>
</tr>
<tr>
<td></td>
<td>Turtle Functions .......................... .................. 205</td>
</tr>
<tr>
<td></td>
<td>Turtle Motion .......................... ........................ 205</td>
</tr>
<tr>
<td></td>
<td>Pen Control ................................................. 207</td>
</tr>
<tr>
<td></td>
<td>Turtle State ................................................ 209</td>
</tr>
<tr>
<td></td>
<td>Event Handling ................................................ 210</td>
</tr>
<tr>
<td></td>
<td>Functions Related to the Window and Canvas ............... 211</td>
</tr>
<tr>
<td></td>
<td>Window Functions ........................................... 211</td>
</tr>
<tr>
<td></td>
<td>Input Functions ............................................. 212</td>
</tr>
</tbody>
</table>
Appendix B  Solutions to Exercises ..................................... 213
  Exercise Solutions for Chapter 1 ........................................ 213
    Exercise 1 ................................................................. 213
    Exercise 2 ................................................................. 213
  Exercise Solutions for Chapter 2 ........................................ 213
    Exercise 1 ................................................................. 213
    Exercise 2 ................................................................. 214
  Exercise Solutions for Chapter 3 ........................................ 216
    Exercise 1 ................................................................. 216
    Exercise 2 ................................................................. 216
  Exercise Solutions for Chapter 4 ........................................ 216
    Exercise 1 ................................................................. 216
    Exercise 2 ................................................................. 217
  Exercise Solutions for Chapter 5 ........................................ 218
    Exercise 1 ................................................................. 218
    Exercise 2 ................................................................. 218
  Exercise Solutions for Chapter 6 ........................................ 219
    Exercise 1 ................................................................. 219
    Exercise 2 ................................................................. 221
  Exercise Solutions for Chapter 7 ........................................ 222
    Exercise 1 ................................................................. 222
    Exercise 2 ................................................................. 222
  Exercise Solutions for Chapter 8 ........................................ 224
    Exercise 1 ................................................................. 224
    Exercise 2 ................................................................. 227
  Exercise Solutions for Chapter 9 ........................................ 229
    Exercise 1 ................................................................. 229
    Exercise 2 ................................................................. 231

Index ................................................................................. 233
Welcome to *Python Programming for Teens*. Whether you’re under 20 or just a teenager at heart, this book will introduce you to computer programming. You can use it in a classroom or on your own. The only assumption is that you know how to use a modern computer system with a keyboard, screen, and mouse.

To make your learning experience fun and interesting, you will write programs that draw pictures on the screen and allow you to interact with them by using the mouse. Along the way, you will learn the basic principles of program design and problem solving with computers. You will then be able to apply these ideas and techniques to solve problems in almost any area of study. But most of all, you will experience the joy of building things that work *and* look great!

**Why Python?**

Computer technology and applications have become increasingly more sophisticated over the past several decades, and so has the computer science curriculum, especially at the introductory level. Today’s students learn a bit of programming and problem solving and are then expected to move quickly into topics like software development, complexity analysis, and data structures that, 20 years ago, were reserved for advanced courses. In addition, the ascent of object-oriented programming as a dominant method has led instructors and textbook authors to bring powerful, industrial-strength programming languages such as C++ and Java into the introductory curriculum. As a result, instead of experiencing the rewards and excitement of computer programming, beginning students
often become overwhelmed by the combined tasks of mastering advanced concepts and learning the syntax of a programming language.

This book uses the Python programming language as a way of making the learning experience manageable and attractive for students and instructors alike. Python offers the following pedagogical benefits:

- Python has simple, conventional syntax. Its statements are close to those of ordinary English, and its expressions use the conventional notation found in algebra. Thus, beginners can spend less time learning the syntax of a programming language and more time learning to solve interesting problems.

- Python has safe semantics. Any expression or statement whose meaning violates the definition of the language produces an error message.

- Python scales well. It is easy for beginners to write simple programs. Python also includes all the advanced features of a modern programming language, such as support for data structures and object-oriented software development, for use when they become necessary.

- Python is highly interactive. Expressions and statements can be entered at an interpreter’s prompts to allow the programmer to try out experimental code and receive immediate feedback. Longer code segments can then be composed and saved in script files to be loaded and run as modules or standalone applications.

- Python is general purpose. In today’s context, this means that the language includes resources for contemporary applications, including media computing and networks.

- Python is free and is in widespread use in the industry. Students can download it to run on a variety of devices. There is a large Python user community, and expertise in Python programming has great resume value.

To summarize these benefits, Python is a comfortable and flexible vehicle for expressing ideas about computation, both for beginners and experts alike. If students learn these ideas well in their first experience with programming, they should have no problems making a quick transition to other languages and technologies needed to achieve their educational or career objectives. Most importantly, beginners will spend less time staring at a computer screen and more time thinking about interesting problems to solve.
Organization of the Book

The approach in this book is easygoing, with each new concept introduced only when you need it.

Chapter 1, “Getting Started with Python,” advises you how to download, install, and start the Python programming software used in this book. You try out simple program commands and become acquainted with the basic features of the Python language that you will use throughout the book.

Chapter 2, “Getting Started with Turtle Graphics,” introduces the basic commands for turtle graphics. You learn to draw pictures with a set of simple commands. Along the way, you discover a thing or two about colors and two-dimensional geometry.

Chapter 3, “Control Structures: Sequencing, Iteration, and Selection,” covers the program commands that allow the computer to make choices and perform repetitive tasks.

Chapters 4, “Composing, Saving, and Running Programs,” shows you how to save your programs in files, so you can give them to others or work on them another day. You learn how to organize a program like an essay, so it is easy for you and others to read, understand, and edit. You also learn a bit about how the computer is able to read, understand, and run a program.

Chapter 5, “Defining Functions,” introduces an important design feature: the function. By organizing your programs with functions, you can simplify complex tasks and eliminate unnecessary duplications in your code.

Chapter 6, “User Interaction with the Mouse and the Keyboard,” covers features that allow people to interact with your programs. You learn program commands for responding to mouse and keyboard events, as well as pop-up dialogs that can take information from your programs’ users.

Chapter 7, “Recursion,” teaches you about another important design strategy called recursion. You write some recursive functions that generate computer art and fractal images.

Chapter 8, “Objects and Classes,” offers a beginner’s guide to the use of objects and classes in programming. You learn how to define new types of objects, such as menu items for choosing colors and grids for board games, and use them in interesting programs.

Chapter 9, “Animations,” concludes the book with a brief introduction to animations. You discover how to get images to move independently and interact in interesting ways.