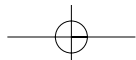
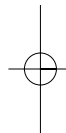
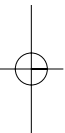


STARTING OUT WITH

Python[®]



STARTING OUT WITH **Python**[®]

Tony Gaddis

Haywood Community College



Boston San Francisco New York
London Toronto Sydney Tokyo Singapore Madrid
Mexico City Munich Paris Cape Town Hong Kong Montreal

Executive Editor	Michael Hirsch
Editorial Assistant	Stephanie Sellinger
Associate Managing Editor	Jeffrey Holcomb
Text Designer	Joyce Cosentino Wells
Cover Designer	Beth Paquin
Photo Research	Beth Anderson
Digital Assets Manager	Marianne Groth
Senior Media Producer	Bethany Tidd
Marketing Manager	Erin Davis
Senior Author Support/ Technology Specialist	Joe Vetere
Senior Manufacturing Buyer	Carol Melville
Senior Media Buyer	Ginny Michaud
Production Coordination	Shelley Creager, Aptara Corp.
Composition and Illustrations	Aptara Corp.
Indexing	Steve Rath

Photo Credits

Cover image © Getty Images / Image Source Pink

Figure 1-3, “The ENIAC computer,” (page 4) is courtesy of U.S. Army Historic Computer Images.

Figure 1-4, “A lab technician holds a modern microprocessor,” (page 4) is courtesy of Intel Corporation.

Figure 1-5, “Memory chips,” (page 5) is courtesy of IBM Corporation.

Rendered art and photographic images in Figures 1-2 (page 3), 1-15 (page 12), 1-16 and 1-17 (page 15), and 1-19 and 1-20 (page 19) © 2007 JUPITERIMAGES and its licensors. All Rights Reserved.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Library of Congress Cataloging-in-Publication Data

Gaddis, Tony.

Starting out with Python / Tony Gaddis.

p. cm.

Includes index.

ISBN-13: 978-0-321-53711-9

ISBN-10: 0-321-53711-4

1. Python (Computer program language) I. Title.

QA76.73.P98G34 2009

005.13'3--dc22

2008001684

Copyright © 2009 Pearson Education, Inc. 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 publisher. Printed in the United States of America. For information on obtaining permission for use of material in this work, please submit a written request to Pearson Education, Inc., Rights and Contracts Department, 501 Boylston Street, Suite 900, Boston, MA 02116, fax (617) 671-3447, or online at <http://www.pearsoned.com/legal/permissions.htm>.

ISBN-13: 978-0-321-53711-9

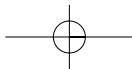
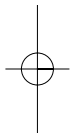
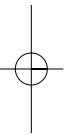
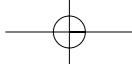
ISBN-10: 0-321-53711-4

1 2 3 4 5 6 7 8 9 10—EB—12 11 10 09 08

Contents at a Glance

Preface xi

Part I:	Programming Fundamentals	
Chapter 1	Introduction to Computers and Programming	1
Chapter 2	Input, Processing, and Output	31
Chapter 3	Simple Functions	77
Chapter 4	Decision Structures and Boolean Logic	113
Chapter 5	Repetition Structures	151
Chapter 6	Value-Returning Functions and Modules	191
Part II:	Using Objects to Perform Tasks	
Chapter 7	Files and Exceptions	225
Chapter 8	Working with Sequences: Strings and Lists	273
Part III:	Object-Oriented Programming	
Chapter 9	Classes and Object-Oriented Programming	329
Chapter 10	Inheritance	373
Part IV:	Advanced Topics	
Chapter 11	Recursion	399
Chapter 12	GUI Programming	419
Appendix A	Installing Python	457
Appendix B	Introduction to IDLE	459
Appendix C	The ASCII Character Set	467
	Index	469
Student CD	The following appendix is on the accompanying Student CD.	
Appendix D	Answers to Checkpoints	



Contents

Preface xi

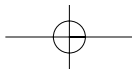
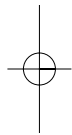
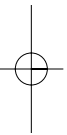
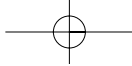
Part I: **Programming Fundamentals**

Chapter 1	Introduction to Computers and Programming	1
1.1	Introduction	1
1.2	Hardware and Software	2
1.3	How Computers Store Data	8
1.4	How a Program Works	13
1.5	Using Python	20
Chapter 2	Input, Processing, and Output	31
2.1	Designing a Program	31
2.2	Input, Processing, and Output	35
2.3	Displaying Output with the <code>print</code> Statement	36
2.4	Comments	39
2.5	Variables	40
2.6	Reading Input from the Keyboard	48
2.7	Performing Calculations	50
2.8	More About Data Output	62
Chapter 3	Simple Functions	77
3.1	Introduction to Functions	77
3.2	Defining and Calling a Function	79
3.3	Designing a Program to Use Functions	85
3.4	Local Variables	91
3.5	Passing Arguments to Functions	93
3.6	Global Variables and Global Constants	103
Chapter 4	Decision Structures and Boolean Logic	113
4.1	The <code>if</code> Statement	113
4.2	The <code>if-else</code> Statement	121
4.3	Comparing Strings	126
4.4	Nested Decision Structures and the <code>if-elif-else</code> Statement	130
4.5	Logical Operators	138
4.6	Boolean Variables	144

viii Contents

Chapter 5	Repetition Structures	151
5.1	Introduction to Repetition Structures	151
5.2	The <code>while</code> Loop: a Condition-Controlled Loop	152
5.3	The <code>for</code> Loop: a Count-Controlled Loop	161
5.4	Calculating a Running Total	173
5.5	Sentinels	176
5.6	Input Validation Loops	179
5.7	Nested Loops	184
Chapter 6	Value-Returning Functions and Modules	191
6.1	Introduction to Value-Returning Functions: Generating Random Numbers	191
6.2	Writing Your Own Functions	200
6.3	The <code>math</code> Module	211
6.4	Storing Functions in Modules	214
Part II:	Using Objects to Perform Tasks	
Chapter 7	Files and Exceptions	225
7.1	Introduction to File Input and Output	225
7.2	Using Loops to Process Files	242
7.3	Processing Records	249
7.4	Exceptions	262
Chapter 8	Working with Sequences: Strings and Lists	273
8.1	Sequences	273
8.2	Working with Strings	273
8.3	Lists	294
Part III:	Object-Oriented Programming	
Chapter 9	Classes and Object-Oriented Programming	329
9.1	Procedural and Object-Oriented Programming	329
9.2	Classes	333
9.3	Working with Instances	350
9.4	Techniques for Designing Classes	358
Chapter 10	Inheritance	373
10.1	Introduction to Inheritance	373
10.2	Polymorphism	388
Part IV:	Advanced Topics	
Chapter 11	Recursion	399
11.1	Introduction to Recursion	399
11.2	Problem Solving with Recursion	402
11.3	Examples of Recursive Algorithms	406

Chapter 12	GUI Programming	419
12.1	Graphical User Interfaces	419
12.2	Using the Tkinter Module	421
12.3	Display Text with Label Widgets	424
12.4	Organizing Widgets with Frames	427
12.5	Button Widgets and Info Dialog Boxes	430
12.6	Getting Input with the Entry Widget	433
12.7	Using Labels as Output Fields	436
12.8	Radio Buttons and Check Buttons	444
Appendix A	Installing Python	457
Appendix B	Introduction to IDLE	459
Appendix C	The ASCII Character Set	467
	Index	469
Student CD	The following appendix is on the accompanying Student CD.	
Appendix D	Answers to Checkpoints	



Preface

Welcome to *Starting Out with Python*. This book uses the Python language to teach programming concepts and problem-solving skills, without assuming any previous programming experience. With easy-to-understand examples, pseudocode, flowcharts, and other tools, the student learns how to design the logic of programs and then implement those programs using Python. This book is ideal for an introductory programming course or a programming logic and design course using Python as the language.

As with all the books in the *Starting Out With* series, the hallmark of this text is its clear, friendly, and easy-to-understand writing. In addition, it is rich in example programs that are concise and practical. The programs in this book include short examples that highlight specific programming topics, as well as more involved examples that focus on problem solving. Each chapter provides one or more case studies that provide step-by-step analysis of a specific problem and shows the student how to solve it.

Control Structures First, Then Classes

Python is a fully object-oriented programming language, but students do not have to understand object-oriented concepts to start programming in Python. This text first introduces the student to the fundamentals of data storage, input and output, control structures, functions, sequences and lists, file I/O, and objects that are created from standard library classes. Then the student learns to write classes, explores the topics of inheritance and polymorphism, and learns to write recursive functions. Finally, the student learns to develop simple event-driven GUI applications.

Brief Overview of Each Chapter

Chapter 1: Introduction to Computers and Programming

This chapter begins by giving a very concrete and easy-to-understand explanation of how computers work, how data is stored and manipulated, and why we write programs in high-level languages. An introduction to Python, interactive mode, script mode, and the IDLE environment is also given.

Chapter 2: Input, Processing, and Output

This chapter introduces the program development cycle, variables, data types, and simple programs that are written as sequence structures. The student learns to write simple programs

that read input from the keyboard, perform mathematical operations, and produce screen output. Pseudocode and flowcharts are also introduced as tools for designing programs.

Chapter 3: Simple Functions

This chapter shows the benefits of modularizing programs and using the top-down design approach. The student learns to define and call simple functions (functions that do not return values), pass arguments to functions, and use local variables. Hierarchy charts are introduced as a design tool.

Chapter 4: Decision Structures and Boolean Logic

In this chapter the student learns about relational operators and Boolean expressions and is shown how to control the flow of a program with decision structures. The `if`, `if-else`, and `if-elif-else` statements are covered. Nested decision structures and logical operators are also discussed.

Chapter 5: Repetition Structures

This chapter shows the student how to create repetition structures using the `while` loop and `for` loop. Counters, accumulators, running totals, and sentinels are discussed, as well as techniques for writing input validation loops.

Chapter 6: Value-Returning Functions and Modules

This chapter begins by discussing common library functions, such as those for generating random numbers. After learning how to call library functions and use their return value, the student learns to define and call his or her own functions. Then the student learns how to use modules to organize functions.

Chapter 7: Files and Exceptions

This chapter introduces sequential file input and output. The student learns to read and write large sets of data and store data as fields and records. The chapter concludes by discussing exceptions and shows the student how to write exception-handling code.

Chapter 8: Working with Sequences: Strings and Lists

This chapter introduces the student to the concept of a sequence in Python and explores the use of two common Python sequences: strings and lists. Several programming techniques are shown using strings with operators, built-in functions, library functions, and string methods. The student also learns to use lists for array-like processing.

Chapter 9: Classes and Object-Oriented Programming

This chapter compares procedural and object-oriented programming practices. It covers the fundamental concepts of classes and objects. Attributes, methods, encapsulation and data hiding, `__init__` functions (which are similar to constructors), accessors, and mutators are discussed. The student learns how to model classes with UML and how to find the classes in a particular problem.

Chapter 10: Inheritance

The study of classes continues in this chapter with the subjects of inheritance and polymorphism. The topics covered include superclasses, subclasses, how `__init__` functions work in inheritance, method overriding, and polymorphism.

Chapter 11: Recursion

This chapter discusses recursion and its use in problem solving. A visual trace of recursive calls is provided and recursive applications are discussed. Recursive algorithms for many tasks are presented, such as finding factorials, finding a greatest common denominator (GCD), and summing a range of values in a list, and the classic Towers of Hanoi example are presented.

Chapter 12: GUI Programming

This chapter discusses the basic aspects of designing a GUI application using the `Tkinter` module in Python. Fundamental widgets, such as labels, button, entry fields, radio buttons, check buttons, and dialog boxes, are covered. The student also learns how events work in a GUI application and how to write callback functions to handle events.

Appendix A: Installing Python

This appendix explains how to install the Python interpreter from the accompanying CD or download it from the Python Web site.

Appendix B: Introduction to IDLE

This appendix gives an overview of the IDLE integrated development environment that comes with Python.

Appendix C: The ASCII Character Set

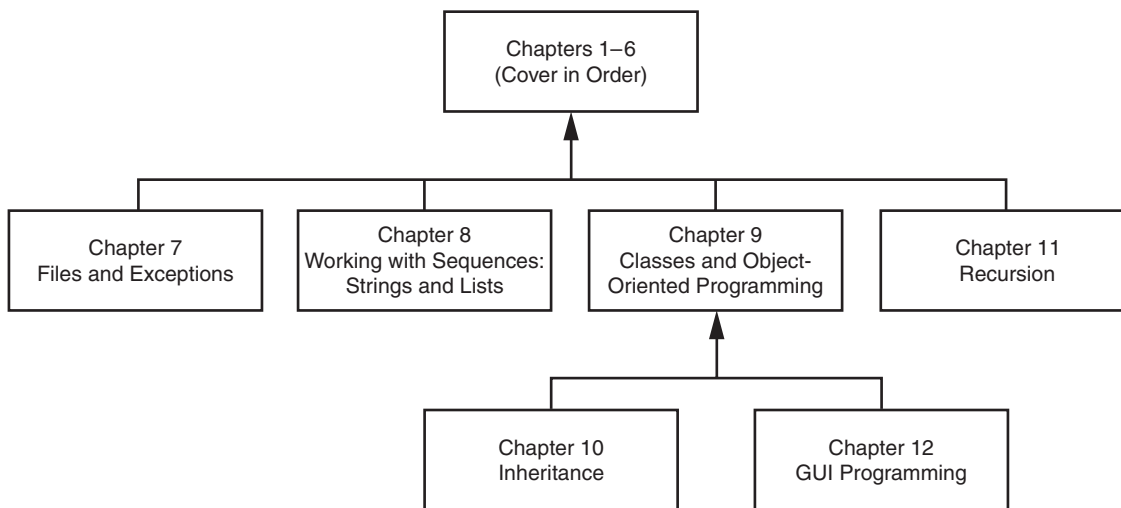
As a reference, this appendix lists the ASCII character set.

Appendix D: Answers to Checkpoint Questions

This appendix gives the answers to the Checkpoint questions that appear throughout the text.

Organization of the Text

The text teaches programming in a step-by-step manner. Each chapter covers a major set of topics and builds knowledge as students progress through the book. Although the chapters can be easily taught in their existing sequence, you do have some flexibility in the order that you wish to cover them. Figure P-1 shows chapter dependencies. Each box represents a chapter or a group of chapters. An arrow points from a chapter to the chapter that must be covered before it.

Figure P-1 Chapter dependencies

Features of the Text

Concept Statements

Each major section of the text starts with a concept statement. This statement concisely summarizes the main point of the section.

Example Programs

Each chapter has an abundant number of complete and partial example programs, each designed to highlight the current topic.

In the Spotlight Case Studies

Each chapter has one or more *In the Spotlight* case studies that provide detailed, step-by-step analysis of problems and show the student how to solve them.

Notes

Notes appear at several places throughout the text. They are short explanations of interesting or often misunderstood points relevant to the topic at hand.

Tips

Tips advise the student on the best techniques for approaching different programming problems.

Warnings

Warnings caution students about programming techniques or practices that can lead to malfunctioning programs or lost data.

Checkpoints

Checkpoints are questions placed at intervals throughout each chapter. They are designed to query the student's knowledge quickly after learning a new topic.

Review Questions

Each chapter presents a thorough and diverse set of review questions and exercises. They include Multiple Choice, True/False, Algorithm Workbench, and Short Answer.

Programming Exercises

Each chapter offers a pool of programming exercises designed to solidify the student's knowledge of the topics currently being studied.

Supplements

Student Resource CD

This CD includes:

- The Python Interpreter, including the IDLE programming environment
- All of the book's example programs
- Appendix D: Answers to Checkpoint Questions

If a CD did not come with your book or you can't locate your CD, visit <http://www.aw.com/cssupport/> to access most of these items.

Instructor Resources

The following supplements are available to qualified instructors only:

- Answers to all of the Review Questions
- Solutions for the exercises
- PowerPoint presentation slides for each chapter
- Test bank

Visit the Addison-Wesley Instructor Resource Center (www.aw.com/irc) or send an email to computing@aw.com for information on how to access them.

Acknowledgments

I want to thank everyone at Addison-Wesley for making the *Starting Out With* series so successful. I am extremely grateful to Michael Hirsch, executive editor, and Stephanie Sellinger, editorial assistant, for guiding me through the process of writing this book. I also want to thank Erin Davis for all of her work as marketing manager. I had a great production team for this book, led by Jeff Holcomb and including Shelley Creager, Brian Baker, David Lindsay (copyeditor), Joyce Cosentino Wells (text design), Beth Paquin (cover design), Bethany Tidd (media), Carol Melville (manufacturing), and Marianne Groth (supplements). Thanks to you all!

Last, but not least, I want to thank my family for all the patience, love, and support they have shown me throughout this and my many other projects.

About the Author

Tony Gaddis is the principal author of the *Starting Out With* series of textbooks. Tony has nearly two decades of experience teaching computer science courses, primarily at Haywood Community College. He is a highly acclaimed instructor who was previously selected as the North Carolina Community College "Teacher of the Year" and has received the Teaching Excellence award from the National Institute for Staff and Organizational Development. The *Starting Out With* series includes introductory books covering C++, Java™, Microsoft® Visual Basic®, Microsoft® C#®, Python®, and Alice, all published by Addison-Wesley. More information about all these books can be found at www.gaddisbooks.com.

