## LOCATION OF VIDEONOTES IN THE TEXT

| Chapter 1 | Creating a World and Adding Objects, p. 19  
Moving the Camera in 3D Space, p. 34  
Creating the Snowy World, p. 54 |
| Chapter 2 | Adding Instructions to an Alice World, p. 62  
Using the Program Design Cycle, p. 82  
Creating Simultaneously Executed Instructions, p. 98  
Creating the Scene from *Alice's Adventures in Wonderland*, p. 109 |
| Chapter 3 | Creating and Using a Variable, p. 116  
Creating a Set Instruction for a Variable, p. 118  
Calling an `ask user` Function, p. 122  
Using Math to Avoid Collisions, p. 132  
Creating the Miles Per Gallon World, p. 146 |
| Chapter 4 | Creating an `If/Else` Instruction, p. 156  
Using a Relational Operator, p. 165  
Testing an Object’s Color Property, p. 169  
Using a `While` Instruction to Make an Object Vanish, p. 182  
Using the `While` Instruction to Move an Object, p. 186  
Creating the Dragon Guardian World, p. 191 |
| Chapter 5 | Creating a Class-Level Method, p. 196  
Passing Arguments to a Method, p. 211  
Writing a Class-Level Function, p. 221  
Creating the Exercise Competition World, p. 243 |
| Chapter 6 | Handling Key Press Events, p. 254  
Handling a Mouse Click Event, p. 261  
Creating the Jumping Fish World, p. 276 |
| Chapter 7 | Creating a List and Using the `For all in order` and `For all together` Instructions, p. 285  
Using the `Let the mouse move <objects>` Event, p. 305  
Creating an Array and a Loop that Steps Through It, p. 311  
Creating the School of Fish World, p. 320 |
| Chapter 8 | Creating a Recursive Method, p. 327  
Creating the Recursive Weightlifting Jock World, p. 339 |
STARTING OUT WITH

Alice

Second Edition
## Brief Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
<td>xii</td>
</tr>
<tr>
<td>Chapter 1</td>
<td>Introduction to Alice and Objects</td>
<td>1</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Programming in Alice</td>
<td>57</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Variables, Functions, Math, and Strings</td>
<td>113</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Decision and Repetition Structures</td>
<td>151</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Methods, Functions, and More about Variables</td>
<td>195</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Events</td>
<td>249</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Lists and Arrays</td>
<td>281</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Recursion</td>
<td>323</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Installing Alice</td>
<td>343</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Answers to Checkpoints</td>
<td>347</td>
</tr>
<tr>
<td>Index</td>
<td></td>
<td>355</td>
</tr>
</tbody>
</table>
Chapter 3  Variables, Functions, Math, and Strings  113

3.1 Variables .................................................113
TUTORIAL 3-1: Creating and using a variable .....................116
TUTORIAL 3-2: Creating a set instruction for a variable ..........118
3.2 Using Functions ..........................................120
TUTORIAL 3-3: Calling an ask user function ....................122
TUTORIAL 3-4: Using a proximity function .....................127
3.3 Creating Math Expressions ..................................130
TUTORIAL 3-5: Using math to avoid collisions .................132
3.4 Working with Strings and Text ................................138
TUTORIAL 3-6: Converting a Number variable to a string ....141

Review Questions ...........................................144

Chapter 4  Decision and Repetition Structures  151

4.1 Boolean Values ..........................................151
4.2 The If/Else Decision Structure ..............................153
TUTORIAL 4-1: Creating an If/Else instruction .................156
4.3 Relational Comparisons and Logical Operators .............163
TUTORIAL 4-2: Using a relational operator ....................165
TUTORIAL 4-3: Testing an object’s color property ..............169
4.4 The Loop Instruction .......................................174
TUTORIAL 4-4: Using the Loop instruction ....................175
4.5 The While Instruction ......................................180
TUTORIAL 4-5: Using a while instruction to make an object vanish ....182
TUTORIAL 4-6: Using the while instruction to move an object ....186

Review Questions ...........................................189

Chapter 5  Methods, Functions, and More about Variables  195

5.1 Writing Custom Class-Level Methods .......................195
TUTORIAL 5-1: Creating a class-level method ..................196
5.2 Saving an Object to a New Class ............................199
TUTORIAL 5-2: Saving an object to a class ....................200
5.3 Stepwise Refinement ......................................203
TUTORIAL 5-3: Completing the WorkOut world ...............207
5.4 Passing Arguments ........................................209
TUTORIAL 5-4: Passing arguments to a method .................211
5.5 Using Class-Level Variables as Properties .................215
TUTORIAL 5-5: Adding a property to an object .................217
5.6 Writing Class-Level Functions .............................220
TUTORIAL 5-6: Writing a class-level function .................221

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.7</td>
<td>World-Level Methods and Variables</td>
<td>226</td>
</tr>
<tr>
<td>5.8</td>
<td>Using Clipboards</td>
<td>229</td>
</tr>
<tr>
<td>5.9</td>
<td>Tips for Visual Effects and Animation</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>Review Questions</td>
<td>241</td>
</tr>
<tr>
<td>6</td>
<td>Events</td>
<td>249</td>
</tr>
<tr>
<td>6.1</td>
<td>Responding to Events</td>
<td>249</td>
</tr>
<tr>
<td>6.2</td>
<td>Handling Key Press and Mouse Events</td>
<td>253</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 6-1: Handling key press events</td>
<td>254</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 6-2: Handling the while a key is pressed event</td>
<td>258</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 6-3: Handling a mouse click event</td>
<td>261</td>
</tr>
<tr>
<td>6.3</td>
<td>Using Events in Simulations and Games</td>
<td>264</td>
</tr>
<tr>
<td>6.4</td>
<td>Tips for Games and Simulations</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td>Review Questions</td>
<td>275</td>
</tr>
<tr>
<td>7</td>
<td>Lists and Arrays</td>
<td>281</td>
</tr>
<tr>
<td>7.1</td>
<td>Lists</td>
<td>281</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 7-1: Creating a list and using the For all in order and For all together instructions</td>
<td>285</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 7-2: More complex list processing</td>
<td>291</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 7-3: Using the Let the mouse move &lt;objects&gt; event</td>
<td>305</td>
</tr>
<tr>
<td>7.2</td>
<td>Arrays</td>
<td>308</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 7-4: Creating an array and a loop that steps through it</td>
<td>311</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 7-5: Randomly selecting an array element</td>
<td>317</td>
</tr>
<tr>
<td></td>
<td>Review Questions</td>
<td>319</td>
</tr>
<tr>
<td>8</td>
<td>Recursion</td>
<td>323</td>
</tr>
<tr>
<td>8.1</td>
<td>Introduction to Recursion</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 8-1: Creating a recursive method</td>
<td>327</td>
</tr>
<tr>
<td>8.2</td>
<td>Problem Solving with Recursion</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 8-2: Recursive problem solving in animation</td>
<td>331</td>
</tr>
<tr>
<td></td>
<td>TUTORIAL 8-3: Writing a recursive mathematical function</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>Review Questions</td>
<td>338</td>
</tr>
<tr>
<td>A</td>
<td>Installing Alice</td>
<td>343</td>
</tr>
<tr>
<td>B</td>
<td>Answers to Checkpoints</td>
<td>347</td>
</tr>
<tr>
<td></td>
<td>Index</td>
<td>355</td>
</tr>
</tbody>
</table>
Preface

This book teaches computer programming using Alice, a revolutionary software system that is freely available from Carnegie Mellon University. Alice is a three-dimensional graphical system that can be used to create animations and computer games. With Alice, students build virtual worlds inhabited by objects from the real world, such as people, animals, cars, airplanes, and more. The virtual worlds that students create, and the objects they place in them, can be programmed to perform actions. While learning to program in Alice, students learn the same fundamentals that are taught with traditional languages such as Java, C++, or Visual Basic.

Changes in the Second Edition

This book’s pedagogy, organization, and clear writing style remain the same as in the previous edition. Many improvements have been made, which are summarized here:

- **Online Video Notes**
  An extensive series of online video notes have been developed to accompany this text. Throughout the book, video note icons alert the student to videos that explain many of the book’s hands-on tutorials. Additionally, one programming exercise at the end of each chapter now has an accompanying video note explaining how to develop the problem’s solution. The videos are available at www.pearsonhighered.com/gaddis.

- **Updated for Alice 2.2**
  This book has been updated for compatibility with Alice 2.2.

- **Exporting an Alice World to Video**
  Chapter 2 discusses the process of exporting an Alice world to video.

- **Expanded Coverage of Decision Structures**
  Chapter 4, Decision and Repetition Structures, has a new section on testing the value of an object’s property with an `If/Else` instruction, a new tutorial on testing an object’s color property.

- **Expanded Coverage of Repetition Structures**
  Chapter 4, Decision and Repetition Structures, has a new tutorial on using a `While` instruction to repeatedly manipulate an object’s opacity property.

Additional Programming Problems
Additional programming problems have been added to Chapters 1, 2, 4, and 5. Many of these new problems are designed to focus on a small set of topics from their chapter, and can be completed in a short period of time.

Why Use Alice to Teach Programming?

With Alice, Abstract Concepts Become Concrete
Many students have trouble learning computer programming because the concepts are so abstract. This is where Alice comes to the rescue! Alice brings abstract concepts down to earth, and makes them more concrete. For example, objects take on the form of physical entities, such as people, buildings, animals, and cars, which can be seen on the screen. When an object’s methods are called, they cause the object to perform actions that can be observed. The student gets the sense that he or she is working with a real, tangible object. This experience prepares the student to understand the more abstract object-oriented principles that he or she will encounter when studying traditional programming languages.

Alice Eliminates Syntax Errors
For many beginning students, learning the syntax of a programming language can be a daunting task. Precious time that should be devoted to learning the fundamentals of programming is often spent tracking down missing semicolons or unbalanced braces. Syntax errors in Alice are never a problem because they never happen! The student builds a program by dragging and dropping tiles into an editor. The tiles represent programming instructions and method calls. When values or expressions are needed for variable assignments or as arguments for methods, pop-up menus appear allowing the student to select the needed value or expression.

Runtime errors can still occur, of course, because the student can use the wrong instruction, or get instructions out of order. But, because syntax is not an issue, the student devotes his or her time to developing and debugging algorithms.

Alice Is Motivating
Perhaps the most compelling reason to use Alice is the fact that students enjoy it so much. Rather than writing programs that perform calculations and display text on the screen, students use Alice to create rich animations and computer games. The time that students spend with Alice is productive because they learn the fundamentals of programming while creating virtual worlds.

Alice in the Classroom
Alice is used in a variety of ways in the classroom, and this text is designed to accommodate all of them. The following are a few examples:
This text can be used with Alice for the first part of an introductory programming course before moving to a traditional programming language. Depending on the amount of time devoted to Alice in such a course, the entire book can be covered, or some of the later chapters can be omitted.

This text can be used with Alice for a brief introduction to programming in a computer concepts course or an introduction to technology course. Later chapters can be omitted to fit the amount of time available.

This text can be used in a semester-long course that uses only Alice to teach programming fundamentals.

This text can be used in short courses or summer programs in which Alice is used to teach programming or virtual world building.

**Brief Overview of Each Chapter**

**Chapter 1 Introduction to Alice and Objects**

Chapter 1 explains what an algorithm is and why we use programming languages. The Alice software is introduced and the student learns about objects and three-dimensional graphical environments.

**Chapter 2 Programming in Alice**

In Chapter 2 the student learns about primitive methods in Alice and how to write instructions that call them. The program design cycle is introduced and the student learns to use it to develop a program.

**Chapter 3 Variables, Functions, Math, and Strings**

Chapter 3 introduces variables and their data types. The student also learns about functions and how to call them. Math expressions are introduced and the student learns to use functions that work with strings.

**Chapter 4 Decision and Repetition Structures**

Chapter 4 first discusses Boolean expressions. Then the student learns to write decision structures using the If/Else instruction, and repetition structures using the Loop and While instructions.

**Chapter 5 Methods, Functions, and More about Variables**

In Chapter 5 the student learns to write class-level methods in objects, create new properties in objects, and save objects as new classes. Stepwise refinement is discussed and the student learns to divide a large problem into several methods.

**Chapter 6 Events**

Chapter 6 introduces event-driven programming. The student learns how to handle various events in Alice, with a special emphasis given to developing games.
Chapter 7 Lists and Arrays
Chapter 7 introduces lists and arrays as data structures. The student learns how to create lists and arrays and how to process items stored in them.

Chapter 8 Recursion
Chapter 8 introduces recursion and discusses how to use recursion in problem solving. Examples that use recursion for animation and for calculations are demonstrated.

Appendix A Installing Alice
Appendix A presents step-by-step instructions for installing the Alice software.

Appendix B Answers to Checkpoints
Appendix B provides the answers to all of the Checkpoint questions that appear throughout each chapter. Students can use these answers to check their own progress as they work through the text.

Features of the Text

Concept Statements
Most major sections of the text start with a concept statement. This statement concisely summarizes the main point of the section.

Example Worlds
The text has an abundant number of complete and partial example worlds, each designed to highlight the topic currently being studied.

Tutorials
Each chapter has several hands-on tutorials that lead the student through the process of developing or completing an Alice world. These tutorials give the student experience performing the tasks discussed in the chapters.

Video Notes
A series of online videos, developed specifically for this book, are available for viewing at www.pearsonhighered.com/gaddis. Icons appear throughout the text alerting the student to videos that accompany specific tutorials and programming problems.

Notes
Notes appear throughout the text. They provide short explanations of interesting or frequently misunderstood points relevant to the topic at hand.
Tips
Tips appear throughout the text and advise the student on the best techniques for approaching different programming or animation problems.

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
A thorough set of multiple choice and short answer questions appear at the end of each chapter.

Exercises
At the end of each chapter, following the review questions, appears a set of exercises for developing Alice worlds. These exercises are designed to solidify the student's knowledge of the topics presented in the chapter.

Supplements

Student Resource CD
This CD includes:
- A copy of the Alice software
- A set of example Alice worlds

If you cannot locate your CD, many of these resources are also available at http://www.aw.com/cssupport

Instructor Resources
The following supplements are available to qualified instructors:
- Completed Alice worlds for the tutorials
- Answers to all of the review questions
- Solutions for the exercises
- PowerPoint presentation slides
- Test bank

Please visit the Addison-Wesley Instructor Resource Center (http://www.aw.com/irc) or send an e-mail to computing@aw.com for information on how to access these resources.
Acknowledgments

There have been many helping hands in the development and publication of this text. I would like to thank the following faculty reviewers:

Carol Buse  
*Amarillo College*

Scott A. Hood  
*Kennebec Valley Community College*

Jim McDonald  
*Monmouth University*

W. Brett McKenzie  
*Roger Williams University*

Laurie J. Patterson  
*University of North Carolina Wilmington*

Charles Payne  
*Northern High School*

Charlotte Young  
*South Plains College*

I would also like to thank everyone at Addison-Wesley for making the *Starting Out With* series so successful. I am extremely fortunate to have Michael Hirsch as my editor, and Stephanie Sellinger as editorial assistant, guiding me through the process of revising this book. I am also fortunate to work with the computer science marketing team at Pearson. They do a great job getting my books out to the academic community. I had a great production team for this book, led by Jeff Holcomb, Managing Editor, and Kayla Smith-Tarbox, Production Project Manager. 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 20 years of experience teaching computer science courses 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 also includes introductory books using *C++*, *Java™*, *Microsoft® Visual Basic®, Microsoft® C#®, Python*, and *Programming Logic and Design*, all published by Addison-Wesley.*