

Welcome to Java Software Solutions for AP\* Computer Science. Specifically designed for beginning programmers, this Advanced Placement book matches the AP Computer Science topic outline and Java subset set forth by the College Board.

## What's new in this Second Edition

This edition was developed to include features of the Java 2 Standard Edition 5.0, known as Java 5, which will be used on the AP Computer Science Exam. It retains all the pedagogy and attention to detail that made the first edition successful. Coverage of Java 5 features includes:

- \* Autoboxing, which automatically converts primitives to wrapper objects and vice-versa, is introduced in section 2.6, and used in examples thereafter.
- \* Type-safe enumeration is introduced in Chapter 2.
- \* The standard `Scanner` class is introduced for keyboard input and parsing in section 2.9. All examples thereafter use `Scanner`.
- \* The enhanced for loop is introduced in Chapter 3 and used in examples thereafter.
- \* Generic types are introduced with `ArrayList` in Chapter 6 and are used in Chapters 9 and 10 when presenting linear and non-linear data structures.
- \* The `Queue`, `Stack`, and `PriorityQueue` classes from the `java.util` package are used, replacing the AP interfaces which are no longer required on the Exam.

Some of the thousands of users of this book have provided excellent feedback that has lead us towards additional enhancements to this edition. These include:

- \* An online “case study” section for each chapter linking the concepts covered in that chapter with the official AP case study. This material is online so that it can change as the Exam case study changes. (See the “Supplements” section of the preface for more information.)
- \* A stronger focus on object-oriented principles throughout the text.
- \* The addition of AP-style Multiple Choice and Free Response Questions at the end of each chapter.

## About this AP\* book

This book has been specifically designed to meet the needs of today’s AP students and teachers. Let’s hit the highlights. Specifically:

- This book is particularly designed for the A exam, though substantial material is presented to help students who will take the AB exam.
- Online sections for each chapter tie the official AP case study in with the concepts covered in that chapter. This material is available on the book’s Web site.
- The discussion and examples fully embrace the Java 2 Standard Edition, Version 5.0.
- We utilize hundreds of example programs, fully implemented for experimentation by students and teachers.
- Graphics are covered in an optional section (called Graphics Track) at the end of each chapter. Building on the excitement of the web, the Graphics Track is intended to further inspire and engage students.
- The Self-Review Questions, Exercises, and Programming Projects have been specifically

designed to adhere to various learning styles, including multiple choice, true/false, and short answer questions.

- The reference material in the appendices includes: Glossary, Unicode Character Set, Java Operators, and the AP Java Class Library.
- A robust ancillary package accompanies this book and is outlined further in this preface.
- The full-color design aids learning by making it easier to distinguish between various elements in code and diagrams. Full-color screen shots make the discussions of graphical interfaces more insightful and realistic.

## Cornerstones of the text

This text is based on the following basic ideas that we believe make for a sound introductory text.

- **True object-orientation.** A text that really teaches a solid object-oriented approach must use what we call object-speak. That is, all processing should be discussed in object-oriented terms. That does not mean, however, that the first program a student sees must discuss the writing of multiple classes and methods. A student should learn to use objects before learning to write them. This text uses a natural progression that culminates in the ability to design real object-oriented solutions.
- **Sound programming practices.** Students should not be taught how to program; they should be taught how to write good software. There's a difference. Writing software is not a set of cookbook actions, and a good program is more than a collection of statements. This text integrates practices that serve as the foundation of good programming skills. These practices are used in all examples and are reinforced in the discussions. Students learn how to solve problems as well as how to implement solutions. We introduce and integrate basic software engineering techniques throughout the text.
- **Examples.** Students learn by example. This text is filled with fully implemented examples that demonstrate specific concepts. We have intertwined small, readily understandable examples with larger, more realistic ones. There is a balance between graphics and nongraphical programs and between applets and applications. Additional examples can be found on the book's Web site.
- **Graphics and GUIs.** Graphics can be a great motivator for students, and their use can serve as excellent examples of object-orientation. As such, we use them throughout the text in a well-defined set of sections that we call the Graphics Track. This coverage includes the use of event processing and graphical user interfaces (GUIs). Students learn to build GUIs in the appropriate way by using a natural progression of topics. The Graphics Track can be avoided entirely for those who do not choose to use graphics.

## chapter breakdown

Chapter 1 (Computer Systems) introduces computer systems in general, including basic architecture and hardware, networking, programming, and language translation. Java is introduced in this chapter, and the basics of program development are discussed. This chapter contains broad introductory material that can be covered while

students become familiar with their development environment.

Chapter 2 (Objects and Primitive Data) establishes the concept of objects and how they can be used. Many predefined classes from the Java standard library are explored and used. Enumerated types, primitive types, operators, and expressions are also explored.

Chapter 3 (Program Statements) covers most of the fundamental statements including conditionals, loops, and iterators. Some additional operators are introduced at this point as well. Establishing key statements at this point allows the classes of the next chapter to be fully functional and realistic.

Chapter 4 (Writing Classes) explores issues related to writing classes and methods. Topics include instance data, visibility, scope, method parameters, and return types. Method overloading is covered as well. Some of the more involved topics are deferred to or revisited in Chapter 5. The key to Chapter 4 is the many fully implemented, realistic classes that are presented as examples of class design.

Chapter 5 (Enhancing Classes) covers additional issues related to class design and revisits topics that need further exploration. Object references are revisited and carefully explored, and their impact on parameter passing is discussed. Exceptions, interfaces, and their effect on design are also covered. Finally, guidelines on designing classes, as well as testing, are given.

Chapter 6 (Arrays) contains extensive coverage of arrays and array processing. Topics include multidimensional arrays, searching, and sorting. The ArrayList class is explored as well, and generic types are introduced here.

Chapter 7 (Inheritance) covers class derivations and associated concepts such as class hierarchies, overriding, and polymorphism. Emphasis is put on the proper use of inheritance and its role in software design.

Chapter 8 (Recursion) covers the concept, implementation, and proper use of recursion. Several examples from various domains are used to demonstrate how recursive techniques make certain types of processing elegant. Recursive sorting algorithms are also covered.

Chapter 9 (Linear Data Structures) introduces the idea of a collection and its underlying data structure. Abstraction is revisited in this context and the linked list, queue, and stack data structures are explored.

Chapter 10 (Non-linear Data Structures) further explores dynamic data structures, including trees and heaps. Sets and maps are also introduced and hashtables are revisited.

## Supplements

Students are welcome to visit [www.aw.com/cssupport](http://www.aw.com/cssupport) (author: Lewis/Loftus/Cocking) for the following resources.

• **Source Code** to all program examples in the text.

\* **Case Study** section for each chapter linking the concepts covered in that chapter with the official AP case study.

The following supplements are available on-line for qualified teachers only. Please contact your representative for information. You can find your rep in the “Find your sales rep” section at [www.phschool.com](http://www.phschool.com).

\* **Lesson Plans with Pacing Guide.**

\* **Solutions to all end of chapter exercises and programming projects.**

- \* PowerPoint slides.
- \* Test Bank with powerful test generator software, includes a wealth of free response, multiple choice, and true/false type questions
- \* Source Code to all program examples in the text.
- \* Lab Manual with solutions

## Acknowledgments

We are most grateful to the teachers and students from around the world who have provided their feedback on previous editions of this book. We are pleased to see the depth of the faculty's concern for their students and the students' thirst for knowledge. Your comments and questions are always welcome.

Michael Hirsch and Lindsey Triebel, our editors at Addison-Wesley, went above and beyond the call of duty to ensure that the book met the highest quality standards. Their support and enthusiasm are greatly appreciated. We are also grateful to Michelle Brown and Dana Lopreato for making sure that teachers understand the pedagogical advantages of this text. The devotion that the Addison-Wesley folks show to their books is evident in the high-quality results.

The production team for this edition is a group of gifted and hard-working people—miracle workers all. Thanks go to Jeff Holcomb, Marilyn Lloyd, Joyce Wells (for the wonderful cover), AND THE ARGOSY PEOPLE. The quality of the book is due largely to their personal attention, and it is greatly appreciated.

Special thanks also go to the following members of the AP community who offered valuable feedback during the development of this edition:

G. Lynne Ryan- North Crowley High School, TX  
 Kim Murphy- Weber High School, UT  
 Neil Ascione-Wayne Valley High School, NJ  
 Kathy Gilbert- Cary-Grove High School, IL  
 Gregory King- Dublin Coffman High School, OH  
 Richard Lamb- Cranbrook High School, MI  
 Brian Ellis- Manheim Township School District, PA

The reviewers of previous editions of this text, as well as many other instructors and friends, have provided valuable feedback as well. They include Jenka Guevara, American School Foundation, Mexico City; Leigh Ann Sudol, Fox Lane High School/New York University, NY; Kathleen Weaver, Hillcrest High School, Dallas, TX; John E. Hanna, Teaneck High School, NJ; Brian G. Scarbeau, Lake Highland Preparatory School, FL; Frances Caruso Wolanczyk, Fieldston School, NY; Mark Hanington, Punahou School, Honolulu, HI; Rose M. Hoffman, Catholic Memorial High School, Waukesha, WI; Kathleen Larson, Kingston High School, Kingston, NY; Bob Beck, Villanova University; Paul Gormley, Villanova University; Cathy Helwig, Villanova University; Dan Joyce, Villanova University; Najib Nadi, Villanova University; Beth Taddei, Villanova University; Barbara Zimmerman, Villanova University; John Simms, Marquette University; Marian Manyo, Marquette University.

Most importantly, we thank our families for their support during the busy process of writing, not to mention the other times.

