

Third AP Edition

Java

Methods

Object-Oriented Programming
and
Data Structures

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: <http://www.skylit.com>
e-mail: sales@skylit.com
support@skylit.com

**Copyright © 2015 by Maria Litvin, Gary Litvin, and
Skylight Publishing**

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 authors and Skylight Publishing.

Library of Congress Control Number: 2014922396

ISBN 978-0-9824775-6-4

* AP and Advanced Placement are registered trademarks of The College Board, which was not involved in the production of and does not endorse this book.

The names of commercially available software and products mentioned in this book are used for identification purposes only and may be trademarks or registered trademarks owned by corporations and other commercial entities. Skylight Publishing and the authors have no affiliation with and disclaim any sponsorship or endorsement by any of these product manufacturers or trademark owners.

Oracle, Java, and Java logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates in the U.S. and other countries.

SCRABBLE® is the registered trademark of HASBRO in the United States and Canada and of J.W. Spear and Sons, PLC, a subsidiary of Mattel, Inc., outside the United States and Canada.

1 2 3 4 5 6 7 8 9 10 20 19 18 17 16 15

Printed in the United States of America

Preface

This book offers a thorough introduction to the concepts and practices of object-oriented programming in Java. It also introduces the most common data structures and related algorithms and their implementations in the Java collections framework.

Chapters 1-14 follow the syllabus of the AP Computer Science in Java course. They will prepare you well for the AP CS exam. Chapters 15-18 on file input and output, graphics, graphical user interfaces, and events handling in Java will give you a better sense of real-world Java programming; this material also makes case studies, labs, and exercises more fun. Chapters 19-26 deal with more advanced data structures and algorithms. Chapter 27, Design Patterns, introduces more intricate aspects of object-oriented design and serves as an introduction to design patterns. The last chapter, Computing in Context, discusses creative, responsible, and ethical computer use.

This edition builds on our earlier books, *Java Methods A & AB: OOP and Data Structures* (Skylight Publishing, 2006), and *Java Methods*, 2nd AP Edition (2011). The AB-level AP CS exam was discontinued by the College Board in 2009, but we have decided to keep the data structures chapters in this book for teachers who continue teaching this material and for students who want to learn it on their own.

In this edition we have dropped all references to the GridWorld case study, which is no longer required for AP CS exams, and replaced GridWorld with our own labs and case studies. We have also rearranged the chapters, introducing Java strings and arrays earlier (Chapters 8 and 9), to enable you to work on activities from the College Board's sample labs if so desired. (These labs are examples only, not required for AP CS exams.) We still introduce objects and classes early (Chapter 4), but the details of classes and class hierarchies are treated a little later (Chapters 10 and 12). We have made the material from the former chapter on algorithms more concrete and placed it closer to the discussion of how to implement algorithms in Java using iterations (Chapter 7) and recursion (Chapter 13). We have added Chapter 28. We have revised and tested the Java code in all the student and teacher files to make sure it works with Java 7 and 8.

The book follows four main threads: Java syntax and style, OOP concepts and techniques, algorithms, and Java libraries. As in the software engineering profession itself, these threads are interwoven into an inseparable braid.

We strive to present the technical details while grounding them in clear explanations of the underlying concepts. OOP has an extensive conceptual layer and complex terminology. Fortunately, many OOP concepts are more straightforward than the terminology makes them appear. Most of the key elements are actually quite intuitive: *objects* (entities that combine data elements and functions), *classes* (definitions of types of objects), *methods* (functions that carry out certain tasks), *instantiation* (creating an object of a particular class), *inheritance* (one class extending the features of another class), *encapsulation* (hiding the implementation details of a class), *polymorphism* (calling the correct methods automatically for specific objects disguised as objects of more generic types), and *event-driven* applications (where the operating system, the user, or events in the program trigger certain actions).

We also emphasize good programming style, an element not mandated by formal Java language specifications but essential for writing readable and professional programs.

Our labs and case studies aim to demonstrate the most appropriate uses of the programming techniques and data structures we cover. OOP is widely believed to facilitate teamwork, software maintenance, and software reuse. While it is not possible for an introductory textbook to present a large-scale real-world project as a case study, the case studies and labs in this book offer a taste of how these OOP benefits can play out in larger projects.

It is not our goal to teach exclusively the material required for the AP CS exam. While we mostly stay within the Java AP subset defined by the College Board for AP CS exams, we also want to give you a solid conceptual foundation and introduce sound software design and development practices. If you are preparing for the AP exam, you'll need to be familiar with the College Board course description and use our review book, *Be Prepared for the AP Computer Science Exam in Java* (Skylight Publishing).

We assume that at least two or three class periods each week will be held in a computer lab with students working independently or in small groups. A set of *Student Files* downloadable from this book's web site contains files for all the case studies, labs, and exercises in the book; a downloadable set of *Teacher Files*, available to teachers only, provides complete solutions to all the labs and exercises.

Still, with all the examples and case studies, we leave a lot of work to you, the student. This is not a *Java-in-n-days* book or an *n-hours-to-complete* book. It is a book for learning essential concepts and technical skills at a comfortable pace, for acquiring a repertoire of techniques and examples to work from, and for consulting as needed when you start writing your own Java programs professionally or for fun.

Working through this book will not make you a Java expert right away, but it will bring you to the level of an entry-level Java programmer with a better than average understanding of the fundamental concepts. Object-oriented programming was originally meant to make software development more accessible to beginners, and *Java Methods* is written in that spirit.

Without further delay, let us begin learning object-oriented programming in Java!



Since our first book came out in 1998, many of our colleagues, too many to name, have become good friends. We are grateful to them for their loyal support, encouragement, and the many things they have taught us over the years.

We thank the students in Maria's AP Computer Science classes for their patience while studying from a draft of this book; they have caught several typos and mistakes and made many useful suggestions.

We are grateful to Doug Vermes from Scarsdale High School in New York for several corrections and helpful suggestions.

Our special thanks to Margaret Litvin for her thorough and thoughtful editing.



On the cover: "Eastern Wind" by Alexandra Rozenman, oil on canvas, 24" by 24", detail. Courtesy of Alexandra Rozenman, www.alexandrarozenman.com.



About the Authors

Maria Litvin has taught computer science and mathematics at Phillips Academy in Andover, Massachusetts, since 1987. She is an Advanced Placement Computer Science exam reader and Table Leader and, as a consultant for The College Board, provides AP training for high school computer science teachers. Maria has received the 1999 Siemens Award for Advanced Placement for Mathematics, Science, and Technology for New England and the 2003 RadioShack National Teacher Award. Prior to joining Phillips Academy, Maria taught computer science at Boston University. Maria is co-author of *C++ for You++: An Introduction to Programming and Computer Science* (1998), which became one of the leading high school textbooks for AP Computer Science courses, and of the earlier editions of the *Java Methods* books. Maria is also the co-author of *Be Prepared for the AP Computer Science Exam in Java* (since 1999) and *Mathematics for the Digital Age and Programming in Python* (Skylight Publishing, 2010). Since 2014, as a Code.org “affiliate,” Maria has trained dozens of New England elementary school teachers in teaching elements of computer science to children in grades K-5.

Gary Litvin has worked in many areas of software development including artificial intelligence, pattern recognition, computer graphics, and neural networks. As founder of Skylight Software, Inc., he developed SKYLIGHTS/GX, one of the first GUI prototyping and development tools for C and C++ programmers. Gary led in the development of several state-of-the-art software products including interactive touch screen development tools, OCR and handwritten character recognition systems, and credit card fraud detection software. He is the co-author of *C++ for You++*, the *Java Methods* series, *Be Prepared for the AP Computer Science Exam in Java*, and *Mathematics for the Digital Age and Programming in Python*.