
Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:
<http://www.springer.com/series/7592>

James T. Streib • Takako Soma

Guide to Java

A Concise Introduction to Programming

 Springer

James T. Streib
Department of Computer Science
Illinois College
Jacksonville, IL, USA

Takako Soma
Department of Computer Science
Illinois College
Jacksonville, IL, USA

Series Editor

Ian Mackie

Advisory Board

Samson Abramsky, University of Oxford, Oxford, UK
Karin Breitman, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
Chris Hankin, Imperial College London, London, UK
Dexter Kozen, Cornell University, Ithaca, USA
Andrew Pitts, University of Cambridge, Cambridge, UK
Hanne Riis Nielson, Technical University of Denmark, Kongens Lyngby, Denmark
Steven Skiena, Stony Brook University, Stony Brook, USA
Iain Stewart, University of Durham, Durham, UK

ISSN 1863-7310

ISSN 2197-1781 (electronic)

Undergraduate Topics in Computer Science

ISBN 978-1-4471-6316-9

ISBN 978-1-4471-6317-6 (eBook)

DOI 10.1007/978-1-4471-6317-6

Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2014931850

© Springer-Verlag London 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Purpose

The purpose of this text is to help the reader learn very quickly how to program using the Java programming language. This is accomplished by concentrating on the fundamentals, providing plenty of illustrations and examples, and using visual contour diagrams to illustrate the object-oriented semantics of the language.

Comparison to Other Texts

There are a number of texts on the Java programming language. Some of these texts provide plenty of examples and are very comprehensive, but unfortunately they sometimes seem to cover too many details, which might make it difficult for a beginning programmer to discern which points are the most relevant. There are also other texts that attempt to provide a shortened introduction to the language, but it seems that these texts might not provide the necessary examples and illustrations and might be better suited for readers who have previous programming experience.

Need

This text attempts to fill the gap between the above two types of books. First, it provides plenty of examples and concentrates primarily on the fundamentals of the Java programming language so that the reader can stay focused on the key concepts. Second, by concentrating on the fundamentals, it allows the text to be more concise and yet still accessible to readers who have no prior programming experience. The result is that the reader can learn the Java programming language very quickly and also have a good foundation to learn more complex topics later.

Features of This Text

This text provides many examples and illustrations. It further has an early introduction to object-oriented programming and uses contour diagrams to illustrate various object-oriented concepts. The contour model was originally developed by John B. Johnson [1]. The model was elaborated on by Organick, Forsythe, and Plummer to illustrate subprograms, parameter passing, and recursion in procedural and functional languages [2]. The model seems quite adaptable to newer programming methodologies such as object-oriented programming as illustrated in a paper by the authors of this text [3]. As discussed in that paper, it was shown that the use of contour diagrams can be an effective tool in helping one learn object-oriented concepts in the Java programming language. By acquiring a good working model of objects, there is less chance of possible misconceptions.

In many paragraphs of the text, questions are asked of the reader to help them interact with the material and think about the subject matter just presented. Hopefully the reader will take a few moments to try to answer these questions on their own before proceeding to the answer that immediately follows. To help further reinforce concepts, each chapter has one or more complete programs to illustrate many of the concepts presented and also to help readers learn how to write programs on their own. In addition, for review and practice, there are summaries and exercises provided at the end of each chapter. Further, in the appendices at the end of the text, there are answers to selected exercises and a glossary of important terms. A summary of these features is listed below:

- Stresses the fundamentals
- Provides many examples and illustrations
- Has an early introduction to objects
- Uses contour diagrams to illustrate object-oriented concepts
- Asks readers questions to help them interact with the material
- Has one or more complete programs in every chapter
- Provides chapter summaries
- Includes exercises at the end of each chapter, with selected answers in an appendix
- Has a glossary of important terms

Overview of the Chapters

This text first allows the reader to understand a simple program with the appropriate input, processing, and output, followed by an early introduction to objects. It then looks at selection and iteration structures followed by more object-oriented concepts. Next, strings and arrays are examined. This is followed by recursion, inheritance and polymorphism, and elementary files. The appendices include information on graphical input/output, exception processing, Javadoc, a glossary, and

answers to selected exercises. Lastly there are references and useful websites and an index. The following provides a brief synopsis of the chapters and appendices:

- Chapter 1 provides an introduction to variables, input/output, and arithmetic operations.
- Chapter 2 introduces objects and contour diagrams.
- Chapter 3 explains selection structures.
- Chapter 4 shows how iteration structures work.
- Chapter 5 revisits object-oriented concepts.
- Chapter 6 introduces string variables and processing.
- Chapter 7 illustrates arrays and array processing.
- Chapter 8 examines recursion.
- Chapter 9 explores inheritance and polymorphism.
- Chapter 10 discusses elementary files.
- Appendix A gives an introduction to graphical input/output.
- Appendix B discusses elementary exception processing.
- Appendix C presents the basics of Javadoc.
- Appendix D lists a glossary of key terms.
- Appendix E provides answers to selected exercises.

Scope

As mentioned previously, this text concentrates on the fundamentals of the Java programming language such as input/output, object-oriented programming, arithmetic and logic instructions, control structures, strings, arrays including elementary sorting and searching, recursion, and files. As a result, it might not cover all the details that are found in some other texts, and if necessary, these topics can be supplemented by the instructor or reader, or covered in a subsequent text and/or second semester course.

Audience

This text is intended primarily for readers who have not had any previous programming experience; however, this does not preclude its use by others who have programmed previously. It can serve as a text in an introductory programming course, as an introduction to a second language in a practicum course, as a supplement in a course on the concepts of programming languages, or as a self-study guide in either academe or industry. Although no prior programming is assumed, it is recommended that readers have the equivalent of an introduction to functions course that includes trigonometry which will help with problem solving and understanding the examples presented in the text.

Acknowledgments

The authors would like to thank the reviewers Mark E. Bollman of Albion College, James W. Chaffee of the University of Iowa, Naomi E. Hahn of Illinois College, Carroll W. Morrow of Augustana College, and Curt M. White of DePaul University. Also, the authors would like to acknowledge the students of Illinois College who have read and used various sections of the text in the classroom. On a personal note, James Streib would like to acknowledge his father William J. Streib for their numerous conversations, and thank his wife Kimberly A. Streib and son Daniel M. Streib for their continued patience. Takako Soma would like to thank her family and friends, near and far.

Note that Java is a registered trademark of Oracle and/or its affiliates and that Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Feedback

The possibility of errors exist in any text, therefore any corrections, comments, or suggestions are welcome and can be sent to the authors via the e-mail addresses below. In addition to copies of the complete programs presented in the text, any significant corrections can be found at the website below.

Website: <http://www.jtstreib.com/GuideJavaProgramming.html>

Illinois College
Jacksonville, IL, USA
October 21, 2013

James T. Streib
james.streib@jtstreib.com
Takako Soma
tsoma@mail.ic.edu

Contents

1	Variables, Input/Output, and Arithmetic	1
1.1	Introduction	1
1.2	Java Skeleton	5
1.3	Variables and Constants	6
1.4	Assignment Statements	10
1.5	Output	13
1.6	Input	20
1.7	Arithmetic Statements	22
1.8	Comments	29
1.9	Program Design	30
1.10	Complete Program: Implementing a Simple Program	33
1.11	Summary	36
1.12	Exercises (Items Marked with an * Have Solutions in Appendix E)	36
2	Objects: An Introduction	39
2.1	Introduction	39
2.2	Classes and Objects	40
2.3	Public and Private Data Members	41
2.4	Value-Returning Methods	42
2.5	Void Methods and Parameters	42
2.6	Creating Objects and Invoking Methods	44
2.7	Contour Diagrams	45
2.8	Constructors	50
2.9	Multiple Objects and Classes	53
2.10	Universal Modeling Language (UML) Class Diagrams	60
2.11	Complete Program: Implementing a Simple Class and Client Program	62
2.12	Summary	63
2.13	Exercises (Items Marked with an * Have Solutions in Appendix E)	65

3	Selection Structures	69
3.1	Introduction	69
3.2	If-Then Structure	69
3.3	If-Then-Else Structure	75
3.4	Nested If Structures	78
3.4.1	If-Then-Else-If Structure	78
3.4.2	If-Then-If Structure	80
3.4.3	Dangling Else Problem	82
3.5	Logical Operators	86
3.6	Case Structure	93
3.7	Complete Programs: Implementing Selection Structures	98
3.7.1	Simple Program	98
3.7.2	Program with Objects	101
3.8	Summary	103
3.9	Exercises (Items Marked with an * Have Solutions in Appendix E)	103
4	Iteration Structures	107
4.1	Introduction	107
4.2	Pretest Indefinite Loop Structure	108
4.2.1	Count-Controlled Indefinite Iteration Structure	109
4.2.2	Sentinel Controlled Loop	116
4.3	Posttest Indefinite Loop Structure	120
4.4	Definite Iteration Loop Structure	124
4.5	Nested Iteration Structures	127
4.6	Potential Problems	129
4.7	Complete Programs: Implementing Iteration Structures	130
4.7.1	Simple Program	131
4.7.2	Program with Objects	133
4.8	Summary	138
4.9	Exercises (Items Marked with an * Have Solutions in Appendix E)	138
5	Objects: Revisited	143
5.1	Sending an Object to a Method	143
5.2	Returning an Object from a Method	146
5.3	Overloaded Constructors and Methods	148
5.4	Use of the Reserved Word <code>this</code>	153
5.5	Class Constants, Variables, and Methods	157
5.5.1	Local, Instance, and Class Constants	157
5.5.2	Local, Instance, and Class Variables	162
5.5.3	Class Methods	165
5.6	Complete Programs: Implementing Objects	167
5.6.1	Program Focusing on Overloaded Methods	167
5.6.2	Program Focusing on Class Data Members and Class Methods	175

5.7	Summary	179
5.8	Exercises (Items Marked with an * Have Solutions in Appendix E)	179
6	Strings	185
6.1	Introduction	185
6.2	String Class	185
6.3	String Concatenation	186
6.4	Methods in String Class	188
6.4.1	The length Method	188
6.4.2	The indexOf Method	188
6.4.3	The substring Method	189
6.4.4	Comparison of Two String Objects	191
6.4.5	The equalsIgnoreCase Method	194
6.4.6	The charAt Method	195
6.5	The toString Method	196
6.6	Complete Program: Implementing String Objects	198
6.7	Summary	200
6.8	Exercises (Items Marked with an * Have Solutions in Appendix E)	201
7	Arrays	203
7.1	Introduction	203
7.2	Array Declaration	203
7.3	Array Access	205
7.4	Input, Output, Simple Processing, and Methods	206
7.4.1	Input	207
7.4.2	Output	210
7.4.3	Simple Processing	211
7.4.4	Passing an Array to and from a Method	212
7.5	Reversing an Array	213
7.6	Searching an Array	218
7.6.1	Sequential Search	218
7.6.2	Binary Search	219
7.6.3	Elementary Analysis	221
7.7	Sorting an Array	221
7.7.1	Simplified Bubble Sort	222
7.7.2	Modified Bubble Sort	224
7.8	Two-Dimensional Arrays	225
7.8.1	Declaration, Creation, and Initialization	226
7.8.2	Input and Output	228
7.8.3	Processing Data	229
7.8.4	Passing a Two-Dimensional Array to and from a Method	232
7.8.5	Asymmetrical Two-Dimensional Arrays	234
7.9	Arrays of Objects	236

7.10	Complete Program: Implementing an Array	238
7.11	Summary	242
7.12	Exercises (Items Marked with an * Have Solutions in Appendix E)	242
8	Recursion	245
8.1	Introduction	245
8.2	The Power Function	245
8.3	Stack Frames	253
8.4	Fibonacci Numbers	254
8.5	Complete Program: Implementing Recursion	264
8.6	Summary	266
8.7	Exercises (Items Marked with an * Have Solutions in Appendix E)	266
9	Objects: Inheritance and Polymorphism	267
9.1	Inheritance	267
9.2	Protected Variables and Methods	276
9.3	Abstract Classes	277
9.4	Polymorphism	278
9.5	Complete Program: Implementing Inheritance and Polymorphism	284
9.6	Summary	288
9.7	Exercises (Items Marked with an * Have Solutions in Appendix E)	289
10	Elementary File Input and Output	293
10.1	Introduction	293
10.2	File Input	293
10.3	File Output	298
10.4	File Input and Output Using an Array	300
10.5	Specifying the File Location	303
10.6	Complete Programs: Implementing File Input and Output	305
	10.6.1 Matrix Multiplication	305
	10.6.2 Sorting Data in a File	307
10.7	Summary	309
10.8	Exercises (Items Marked with an * Have Solutions in Appendix E)	309
Appendix A	Simple Graphical Input and Output	311
A.1	Message Dialog Boxes	311
A.2	Input Dialog Boxes	312
A.3	Converting String Input from Input Dialog Boxes to Numbers	314
A.4	Confirmation Dialog Boxes	316
A.5	Option Dialog Boxes	317

Appendix B Exceptions	321
B.1 Exception Class and Error Class	321
B.2 Handling an Exception	322
B.3 Throwing Exceptions and Multiple <code>catch</code> Blocks	325
B.4 Checked and Unchecked Exceptions	330
Appendix C Javadoc Comments	335
C.1 Javadoc	335
C.2 More Javadoc Tags	338
C.3 Generating Javadoc Documentation from a Command Line	339
Appendix D Glossary	341
Appendix E Answers to Selected Exercises	345
References and Useful Websites	353
Index	355