
Texts in Computer Science

Series editors

David Gries, Department of Computer Science, Cornell University, Ithaca, NY,
USA

Orit Hazzan, Faculty of Education in Science and Technology, Technion—Israel
Institute of Technology, Haifa, Israel

More information about this series at <http://www.springer.com/series/3191>

Quentin Charatan • Aaron Kans

Java in Two Semesters

Featuring JavaFX

Fourth Edition

Quentin Charatan
University of East London
London, UK

Aaron Kans
University of East London
London, UK

ISSN 1868-0941

Texts in Computer Science

ISBN 978-3-319-99419-2

<https://doi.org/10.1007/978-3-319-99420-8>

ISSN 1868-095X (electronic)

ISBN 978-3-319-99420-8 (eBook)

Library of Congress Control Number: 2018961214

1st edition: © The McGraw-Hill Companies 2002

2nd edition: © McGraw-Hill Education (UK) Limited 2006

3rd edition: © McGraw-Hill Education (UK) Limited 2009

4th edition: © Springer Nature Switzerland AG 2019

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.

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.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Alexi

Quentin Charatan

To Wendy

Aaron Kans

Preface

Accompanying Web site: <https://www.springer.com/book/9783319994192>.

As with previous editions, this book is designed for university students taking a first module in software development or programming, followed by a second, more advanced module. This book uses Java as the vehicle for the teaching of programming concepts—design concepts are explained using the UML notation. The topic is taught from first principles and assumes no prior knowledge of the subject.

This book is organized so as to support two twelve-week, one-semester modules, which might typically comprise a two-hour lecture, a one-hour seminar, and a one- or two-hour laboratory session. The outcomes at the start of each chapter highlight its key learning objectives, the self-test questions at the end of each chapter ensure that the learning objectives for that chapter have been met, while the programming exercises that follow allow these learning objectives to be applied to complete programs. In addition to these exercises and questions, a case study is developed in each semester to illustrate the use of the techniques covered in the text to develop a non-trivial application. Lecturers who teach on modules that run for fewer than twelve weeks in a semester could treat these case studies as a self-directed student learning experience, rather than as taught topics.

The approach taken in this book is ideal for all students including those entering university with little or no background in the subject matter, perhaps coming from pre-degree courses in other disciplines, or perhaps returning to study after long periods away from formal education. It is the authors' experience that such students have enormous difficulties in grasping the fundamental programming concepts the first time round and therefore require a simpler and gentler introduction to the subject that is presented in most standard texts.

This book takes an integrated approach to software development by covering such topics as basic design principles and standards, testing methodologies, and the user interface, as well as looking at detailed implementation topics.

In the first semester, considerable time is spent concentrating on the fundamental programming concepts such as declarations of variables and basic control structures, methods and arrays, prior to introducing students to classes and objects, inheritance, graphics, and event-driven programming.

The second semester covers more advanced topics such as interfaces, exceptions, collection classes from the Java collections framework, advanced graphics,

file-handling techniques, packages, the implementation of multi-threaded programs, socket programming, and processing collections using streams.

The fourth edition achieves three main goals. Firstly, it incorporates all the very useful feedback on the third edition that we have received from students and lecturers since its publication. Secondly, it includes many new questions and programming exercises at the end of the chapters. Finally, it includes new material to bring it completely up to date with the current developments in the field—in particular a number of key developments that were introduced in Java 8 which, according to OracleTM, is “the most significant re-engineering of the language since its launch.”

One key feature of this new edition is that all graphical user interface developments are based on *JavaFX*, rather than the Swing Technology used in previous editions. JavaFX allows for the creation of sophisticated modern graphical interfaces that can run on a variety of devices and is now Oracle’s preferred technology for building such interfaces, having decided that Swing will no longer be developed. JavaFX therefore plays a very significant role throughout the new text, and three new chapters are devoted to it.

Other key developments arising from Java 8 that have been incorporated into the new text include *lambda expressions*, which allow us to simplify development considerably by passing functions as arguments to methods, and the new *Stream API*, a technology that allows us to process collections in a very concise, declarative style of programming.

In addition to the above key changes, we also introduce techniques to improve the robustness of code—in particular the *Optional* class for dealing with empty values and the *try-with-resources* construct to ensure resources such as files are safely closed before exiting methods.

As well as adding these new features, some existing chapters have undergone significant enhancements. The Java Collections Framework chapter, for example, has been expanded to include a comprehensive section on the *sort* methods available in various classes and interfaces in Java. The coverage of generics has also been considerably expanded and the packages chapter now introduces the *Hibernate ORM* technology for accessing remote databases.

The accompanying Web site (see URL above) contains all the codes from the textbook and a guide on how to install and use the NetBeansTM Java IDE, as well as a collection of other useful resources.

We would like to thank our publisher, Springer, for the encouragement and guidance that we have received throughout the production of this book. Additionally, we would especially like to thank the computing students of the University of East London for their thoughtful comments and feedback and Steven Martin for his help and advice. For support and inspiration, special thanks are due once again to our families and friends.

London, UK

Quentin Charatan
Aaron Kans

Contents

Part I Semester One

1	The First Step	3
1.1	Introduction	3
1.2	Software	4
1.3	Compiling Programs	4
1.4	Programming in Java	5
1.5	Integrated Development Environments (IDEs)	6
1.6	Java Applications	8
1.7	Your First Program	10
	1.7.1 Analysis of the “Hello World” Program	11
	1.7.2 Adding Comments to a Program	13
1.8	Output in Java	14
1.9	Self-test Questions	16
1.10	Programming Exercises	17
2	Building Blocks	19
2.1	Introduction	19
2.2	Simple Data Types in Java	19
2.3	Declaring Variables in Java	21
2.4	Assignments in Java	23
2.5	Creating Constants	25
2.6	Arithmetic Operators	25
2.7	Expressions in Java	27
2.8	More About Output	30
2.9	Input in Java: The <i>Scanner</i> Class	31
2.10	Program Design	35
2.11	Self-test Questions	36
2.12	Programming Exercises	38
3	Selection	41
3.1	Introduction	41
3.2	Making Choices	42

3.3	The ‘if’ Statement	43
3.3.1	Comparison Operators	46
3.3.2	Multiple Instructions Within an ‘if’ Statement	47
3.4	The ‘if...else’ Statement	49
3.5	Logical Operators	51
3.6	Nested ‘if...else’ Statements	53
3.7	The ‘switch’ Statement	55
3.7.1	Grouping Case Statements	56
3.7.2	Removing Break Statements	57
3.8	Self-test Questions	59
3.9	Programming Exercises	61
4	Iteration	65
4.1	Introduction	65
4.2	The ‘for’ Loop	67
4.2.1	Varying the Loop Counter	70
4.2.2	The Body of the Loop	72
4.2.3	Revisiting the Loop Counter	76
4.3	The ‘while’ Loop	77
4.4	The ‘do...while’ Loop	79
4.5	Picking the Right Loop	83
4.6	The ‘break’ Statement	84
4.7	The ‘continue’ Statement	86
4.8	Self-test Questions	88
4.9	Programming Exercises	91
5	Methods	95
5.1	Introduction	95
5.2	Declaring and Defining Methods	96
5.3	Calling a Method	98
5.4	Method Input and Output	99
5.5	More Examples of Methods	103
5.6	Variable Scope	107
5.7	Method Overloading	109
5.8	Using Methods in Menu-Driven Programs	112
5.9	Self-test Questions	115
5.10	Programming Exercises	117
6	Arrays	119
6.1	Introduction	119
6.2	Creating an Array	120
6.3	Accessing Array Elements	124
6.4	Passing Arrays as Parameters	129
6.5	Varargs	131

6.6	Returning an Array from a Method	134
6.7	The Enhanced ‘for’ Loop	137
6.8	Some Useful Array Methods	139
6.8.1	Array Maximum	139
6.8.2	Array Summation	141
6.8.3	Array Membership	141
6.8.4	Array Search	142
6.8.5	The Final Program	143
6.9	Multi-dimensional Arrays	148
6.9.1	Creating a Two-Dimensional Array	148
6.9.2	Initializing Two-Dimensional Arrays	149
6.9.3	Processing Two-Dimensional Arrays	150
6.9.4	The <i>MonthlyTemperatures</i> Program	151
6.10	Ragged Arrays	155
6.11	Self-test Questions	158
6.12	Programming Exercises	161
7	Classes and Objects	163
7.1	Introduction	163
7.2	Classes as Data Types	163
7.3	Objects	165
7.4	The <i>Oblong</i> Class	166
7.5	The <i>OblongTester</i> Program	171
7.6	Strings	173
7.6.1	Obtaining Strings from the Keyboard	173
7.6.2	The Methods of the <i>String</i> Class	174
7.6.3	Comparing Strings	176
7.6.4	Entering Strings Containing Spaces	178
7.7	Our Own <i>Scanner</i> Class for Keyboard Input	179
7.8	The <i>Console</i> Class	181
7.9	The <i>BankAccount</i> Class	182
7.10	Arrays of Objects	185
7.11	The <i>ArrayList</i> Class	188
7.12	Self-test Questions	190
7.13	Programming Exercises	192
8	Implementing Classes	195
8.1	Introduction	195
8.2	Designing Classes in UML Notation	196
8.3	Implementing Classes in Java	198
8.3.1	The <i>Oblong</i> Class	198
8.3.2	The <i>BankAccount</i> Class	202
8.4	The <i>static</i> Keyword	205
8.5	Initializing Attributes	208

8.6	The <i>EasyScanner</i> Class	209
8.7	Passing Objects as Parameters	209
8.8	Collection Classes	211
8.8.1	The <i>Bank</i> Class	211
8.8.2	Testing the <i>Bank</i> Class	217
8.9	The Benefits of Object-Oriented Programming	223
8.10	Self-test Questions	223
8.11	Programming Exercises	227
9	Inheritance	235
9.1	Introduction	235
9.2	Defining Inheritance	236
9.3	Implementing Inheritance in Java	237
9.4	Extending the <i>Oblong</i> Class	241
9.5	Method Overriding	245
9.6	Abstract Classes	250
9.7	Abstract Methods	253
9.8	The <i>final</i> Modifier	257
9.9	The <i>Object</i> Class	257
9.10	The <i>toString</i> Method	258
9.11	Wrapper Classes and Autoboxing	259
9.12	Self-test Questions	260
9.13	Programming Exercises	264
10	Introducing JavaFX	265
10.1	Introduction	265
10.2	A Brief History of Java Graphics	266
10.3	JavaFX: An Overview	267
10.4	2D Graphics: The <i>SmileyFace</i> Class	269
10.5	Event-Handling in JavaFX: The <i>ChangingFace</i> Class	275
10.6	Some More 2D Shapes	281
10.7	An Interactive Graphics Class	282
10.8	A Graphical User Interface (GUI) for the <i>Oblong</i> Class	285
10.9	Containers and Layouts	288
10.9.1	More About <i>HBox</i> and <i>VBox</i>	288
10.9.2	<i>GridPane</i>	290
10.9.3	<i>StackPane</i>	291
10.9.4	<i>FlowPane</i> and <i>BorderPane</i>	292
10.10	Borders, Fonts and Colours	293
10.10.1	Borders	294
10.10.2	Fonts	295
10.10.3	Colours	296
10.11	Number Formatting	297
10.12	A Metric Converter	299

10.13	Self-test Questions	302
10.14	Programming Exercises	303
11	Case Study—Part 1	307
11.1	Introduction	307
11.2	The Requirements Specification	308
11.3	The Design	308
11.4	Implementing the <i>Payment</i> Class	310
11.5	The <i>PaymentList</i> Class	313
11.5.1	Javadoc	317
11.5.2	Code Layout	319
11.6	Testing the <i>PaymentList</i> Class	320
11.7	Implementing the <i>Tenant</i> Class	328
11.8	Implementing the <i>TenantList</i> Class	330
11.9	Self-test Questions	334
11.10	Programming Exercises	334
12	Case Study—Part 2	335
12.1	Introduction	335
12.2	Keeping Permanent Records	335
12.3	Design of the <i>Hostel</i> Class	336
12.4	Design of the GUI	338
12.5	Designing the Event-Handlers	341
12.6	Implementing the <i>Hostel</i> Class	344
12.7	Testing the System	350
12.8	What Next?	353
12.9	Self-test Questions	354
12.10	Programming Exercises	354
Part II Semester Two		
13	Interfaces and Lambda Expressions	357
13.1	Introduction	357
13.2	An Example	358
13.3	Interfaces	359
13.4	Inner Classes	364
13.5	Anonymous Classes	364
13.6	Lambda Expressions	368
13.6.1	The Syntax of Lambda Expressions	369
13.6.2	Variable Scope	371
13.6.3	Example Programs	371
13.6.4	Method References—The Double Colon Operator	374
13.7	Generics	376

13.7.1	Bounded Type Parameters	379
13.7.2	Wildcards	382
13.8	Other Interfaces Provided with the Java Libraries	383
13.9	Polymorphism and Polymorphic Types	385
13.9.1	Operator Overloading	385
13.9.2	Method Overloading	385
13.9.3	Method Overriding	385
13.9.4	Type Polymorphism	386
13.10	Self-test Questions	386
13.11	Programming Exercises	391
14	Exceptions	393
14.1	Introduction	393
14.2	Pre-defined Exception Classes in Java	394
14.3	Handling Exceptions	395
14.3.1	Claiming an Exception	398
14.3.2	Catching an Exception	401
14.4	The ‘finally’ Clause	403
14.5	The ‘Try-with-Resources’ Construct	406
14.6	Null-Pointer Exceptions	408
14.7	The <i>Optional</i> Class	410
14.8	Exceptions in GUI Applications	413
14.9	Using Exceptions in Your Own Classes	416
14.9.1	Throwing Exceptions	417
14.9.2	Creating Your Own Exception Classes	419
14.10	Documenting Exceptions	421
14.11	Self-test Questions	422
14.12	Programming Exercises	423
15	The Java Collections Framework	427
15.1	Introduction	427
15.2	The <i>List</i> Interface and the <i>ArrayList</i> Class	428
15.2.1	Creating an <i>ArrayList</i> Collection Object	429
15.2.2	The Interface Type Versus the Implementation Type	430
15.2.3	<i>List</i> Methods	431
15.3	The Enhanced <i>for</i> Loop and Java Collections	434
15.4	The <i>forEach</i> Loop	435
15.5	The <i>Set</i> Interface and the <i>HashSet</i> Class	436
15.5.1	<i>Set</i> Methods	437
15.5.2	Iterating Through the Elements of a Set	438
15.5.3	<i>Iterator</i> Objects	440

15.6	The <i>Map</i> Interface and the <i>HashMap</i> Class	443
15.6.1	<i>Map</i> Methods	444
15.6.2	Iterating Through the Elements of a Map	446
15.7	Using Your Own Classes with Java's Collection Classes	447
15.7.1	The <i>Book</i> Class	448
15.7.2	Defining an <i>equals</i> Method	450
15.7.3	Defining a <i>hashCode</i> Method	450
15.7.4	The Updated <i>Book</i> Class	452
15.8	Developing a Collection Class for <i>Book</i> Objects	453
15.9	Sorting Objects in a Collection	456
15.9.1	The <i>Collections.sort</i> and <i>Arrays.sort</i> Methods	456
15.9.2	The <i>Comparable<T></i> Interface	458
15.9.3	The <i>Comparator<T></i> Interface	459
15.10	Self-test Questions	463
15.11	Programming Exercises	465
16	Advanced JavaFX	469
16.1	Introduction	469
16.2	Input Events	470
16.2.1	Mouse Events	470
16.2.2	Key Events	473
16.3	Binding Properties	477
16.4	The <i>Slider</i> Class	479
16.5	Multimedia Nodes	482
16.5.1	Embedding Images	483
16.5.2	Embedding Videos	486
16.5.3	Embedding Web Pages	489
16.6	Cascading Style Sheets	491
16.7	Self-test Questions	496
16.8	Programming Exercises	497
17	JavaFX: Interacting with the User	499
17.1	Introduction	499
17.2	Drop-Down Menus	500
17.3	Context (Pop-Up) Menus	503
17.4	Combo Boxes	507
17.5	Check Boxes and Radio Buttons	509
17.6	A Card Menu	513
17.7	The <i>Dialog</i> Class	518
17.8	Self-test Questions	524
17.9	Programming Exercises	525

18	Working with Files	527
18.1	Introduction	527
18.2	Input and Output	528
18.3	Input and Output Devices	528
18.4	File-Handling	530
18.4.1	Encoding	530
18.4.2	Access	531
18.5	Reading and Writing to Text Files	531
18.6	Reading and Writing to Binary Files	539
18.7	Reading a Text File Character by Character	541
18.8	Object Serialization	542
18.9	Random Access Files	544
18.10	Self-test Questions	549
18.11	Programming Exercises	550
19	Packages	553
19.1	Introduction	553
19.2	Understanding Packages	553
19.3	Accessing Classes in Packages	555
19.4	Developing Your Own Packages	558
19.5	Package Scope	559
19.6	Running Applications from the Command Line	560
19.7	Deploying Your Packages	563
19.8	Adding External Libraries	564
19.8.1	Accessing Databases Using JDBC	564
19.8.2	Accessing Databases Using Hibernate	568
19.9	Self-test Questions	574
19.10	Programming Exercises	575
20	Multi-threaded Programs	577
20.1	Introduction	577
20.2	Concurrent Processes	578
20.3	Threads	578
20.4	The <i>Thread</i> Class	580
20.5	Thread Execution and Scheduling	582
20.6	Synchronizing Threads	584
20.7	Thread States	585
20.8	Multithreading and JavaFX	587
20.8.1	The <i>Task</i> Class	587
20.8.2	The <i>Service</i> Class	590
20.8.3	Automating the <i>ChangingFace</i> Application	591
20.8.4	Running a Task in the Background	594
20.8.5	Animation Using a Series of Images	596

20.9	Self-test Questions	599
20.10	Programming Exercises	600
21	Advanced Case Study	603
21.1	Introduction	603
21.2	System Overview	604
21.3	Requirements Analysis and Specification	604
21.4	Design	606
21.5	Enumerated Types in UML	608
21.6	Implementation	609
21.6.1	Implementing Enumerated Types in Java	609
21.6.2	The <i>Runway</i> Class	611
21.6.3	The <i>Plane</i> Class	612
21.6.4	The <i>Airport</i> Class	616
21.7	Testing	624
21.8	Design of the JavaFX Interface	625
21.9	The <i>TabPane</i> Class	626
21.10	The <i>AirportFrame</i> Class	628
21.11	Self-test Questions	638
21.12	Programming Exercises	639
22	The Stream API	641
22.1	Introduction	641
22.2	Streams Versus Iterations: Example Program	643
22.3	Creating Streams	646
22.4	Intermediate Operations	648
22.5	Operations for Terminating Streams	652
22.5.1	More Examples	652
22.5.2	Collecting Results	654
22.6	Concatenating Streams	656
22.7	Infinite Streams	656
22.8	Stateless and Stateful Operations	657
22.9	Parallelism	658
22.10	Self-test Questions	659
22.11	Programming Exercises	659
23	Working with Sockets	661
23.1	Introduction	661
23.2	Sockets	662
23.3	A Simple Server Application	663
23.4	A Simple Client Application	668
23.5	Connections from Multiple Clients	673

23.6	A Client–Server Chat Application	676
23.7	Self-test Questions	686
23.8	Programming Exercises	686
24	Java in Context	689
24.1	Introduction	689
24.2	Language Size	690
	24.2.1 Pointers	690
	24.2.2 Multiple Inheritance	691
24.3	Language Reliability	694
	24.3.1 Aliasing	695
	24.3.2 Overriding the <i>clone</i> Method	697
	24.3.3 Immutable Objects	700
	24.3.4 Using the <i>clone</i> Method of the <i>Object</i> Class	701
	24.3.5 Copy Constructors	703
	24.3.6 Garbage Collection	704
24.4	The Role of Java	706
24.5	What Next?	706
24.6	Self-test Questions	707
24.7	Programming Exercises	708
	Index	709