



Interactive Object-Oriented Programming in Java

Learn and Test Your Programming Skills

—

Second Edition

—

Vaskaran Sarcar

Foreword by Avirup Mullick

Apress®

Interactive Object-Oriented Programming in Java

**Learn and Test Your
Programming Skills**

Second Edition

Vaskaran Sarcar

Foreword by Avirup Mullick

Apress®

Interactive Object-Oriented Programming in Java

Vaskaran Sarcar
Bangalore, Karnataka, India

ISBN-13 (pbk): 978-1-4842-5403-5
<https://doi.org/10.1007/978-1-4842-5404-2>

ISBN-13 (electronic): 978-1-4842-5404-2

Copyright © 2020 by Vaskaran Sarcar

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.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

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.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: Mathew Moodie
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, email orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please email rights@apress.com or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5403-5. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Dear Reader,

*You motivate me with your nice and loving comments,
you hurt me with your extremely critical comments, but in the
end you help me to become a better person and a better author.*

So, this book is dedicated to you.

Table of Contents

About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Preface	xxv
Who Is This Book For?	xxix
Guidelines for Using This Book	xxxi
Conventions Used in This Book	xxxiii
Foreword	xxxv
Part I: Fundamentals of Object-Oriented Programming	1
Chapter 1: Object-Oriented Programming Concepts	3
Class and Objects	5
Encapsulation	6
Abstraction	6
Inheritance	7
Polymorphism	7
Q&A Session	8
Summary	9
Chapter 2: The Building Blocks: Class and Objects	11
Class	11
Object	11
Constructor	14
Demonstration 1	17
Q&A Session	19

TABLE OF CONTENTS

- Demonstration 2 19
- Analysis 20
- Demonstration 3 26
- Demonstration 4 29
- Demonstration 5 31
- Passing Variable-Length Arguments to Methods 32
 - Demonstration 6 32
 - Analysis 33
 - Q&A Session 33
- Summary..... 35
- Chapter 3: Classes and Objects in Depth..... 37**
 - Static Variables and Methods..... 37
 - Demonstration 1 38
 - Q&A Session 39
 - Access Control 39
 - Demonstration 2 40
 - Getter-Setter Methods..... 42
 - Demonstration 3 42
 - Q&A Session 43
 - Initialization Block..... 43
 - Demonstration 4 44
 - Q&A Session 45
 - Nested Class 47
 - Demonstration 5 47
 - Q&A Session 49
 - Copying an Object 49
 - Using Copy Constructors 49
 - Demonstration 6 50
 - Q&A Session 51
 - Wrapper Class..... 51
 - Demonstration 7 52

Garbage Collection	54
Q&A Session	55
Demonstration 8	57
Finalization.....	60
Demonstration 9	61
Summary.....	63
Chapter 4: The Concept of Inheritance	65
Types of Inheritance.....	65
Single Inheritance.....	65
Hierarchical Inheritance	66
Multi-level Inheritance.....	67
Multiple Inheritance.....	68
Demonstration 1	69
Q&A Session	70
Demonstration 2	70
Q&A Session	72
Demonstration 3	72
Q&A Session	74
Demonstration 4	75
Q&A Session	76
A Special Keyword: super	76
Demonstration 5	77
Demonstration 6	81
Q&A Session	83
Demonstration 7	84
Q&A Session	85
Demonstration 8	86
Q&A Session	87
Demonstration 9	88
Q&A Session	89
Summary.....	90

TABLE OF CONTENTS

- Chapter 5: Get Familiar with Polymorphism..... 91**
- Method Overloading..... 91
 - Demonstration 1 91
 - Q&A Session 93
 - Demonstration 2 95
 - Q&A Session 96
 - Demonstration 3 96
 - Q&A Session 97
 - Demonstration 4 97
 - Q&A Session 98
 - Demonstration 5 99
- Method Overriding 101
 - Demonstration 6 102
 - Q&A Session 104
 - Demonstration 7 106
- Runtime Polymorphism..... 108
 - Demonstration 8 109
 - Q&A Session 111
 - Demonstration 9 111
 - Q&A Session 114
 - Demonstration 10 115
- Using the final Keyword 117
 - Blank final Variables 119
 - Q&A Session 120
 - Demonstration 11 120
 - Q&A Session 122
 - Demonstration 12 124
 - Q&A Session 126
 - Demonstration 13 127
 - Q&A Session 128
 - Demonstration 14 129

Covariant Return Type	130
Demonstration 15	130
Demonstration 16	133
Q&A Session	134
Use of Private Constructors	135
Q&A Session	135
Demonstration 17	136
Q&A Session	138
Summary.....	139
Chapter 6: Abstract Classes and Interfaces: The True Art in OOP	141
Abstract Classes	141
Demonstration 1	142
Demonstration 2	143
Q&A Session	146
Interfaces	151
Demonstration 3	153
Q&A Session	154
Demonstration 4	155
Q&A Session	156
Demonstration 5	157
Q&A Session	158
Demonstration 6	158
Q&A Session	160
Demonstration 7	160
Q&A Session	163
Marker Interface	163
A Quick Tour with Annotations	164
Demonstration 8	165
Demonstration 9	169
Javadoc Snapshots.....	170
Q&A Session	171

TABLE OF CONTENTS

- Default Methods in Interfaces 173
 - Demonstration 10 173
 - Q&A Session 175
 - Demonstration 11 176
 - Q&A Session 178
 - Demonstration 12 179
- Summary 180
- Chapter 7: Packages 183**
 - Creating a Package 185
 - Demonstration 1 188
 - Key Notes About Packages in Java 192
 - Q&A Session 193
 - Demonstration 2 195
 - Q&A Session 196
 - Troubleshooting Common Errors in Command-line Environment 197
 - Summary 200
- Chapter 8: Understanding Class Variables and Class Methods 201**
 - Class Variables and Class Methods 202
 - Demonstration 1 202
 - Working with Nested Classes 203
 - Demonstration 2 203
 - Q&A Session 207
 - Initialization Blocks Versus Constructors 208
 - Demonstration 3 210
 - Method Hiding Versus Method Overriding 214
 - Demonstration 4 215
 - Q&A Session 217
 - Method Overloading 217
 - Demonstration 5 217
 - Q&A Session 220

Static Methods in Interfaces	220
Demonstration 6	221
Summary.....	223
Chapter 9: Quick Recap of OOP Principles.....	225
Q&A Session.....	226
Revisiting the Diamond Problem.....	229
Demonstration 1	230
Q&A Session.....	232
Summary.....	235
Part II: Get Familiar with Advanced Programming	237
Chapter 10: Managing Exceptions	239
Types of Mistakes	239
Definition of Exception	241
Demonstration 1	241
Key Points of the Exception-handling Mechanism	242
Demonstration 2	245
Q&A Session	247
Demonstration 3	247
Multiple catch Clauses.....	250
Q&A Session	253
Demonstration 4	254
Q&A Session	255
Throwing an Exception	255
Demonstration 5	256
Rethrowing an Exception	257
Demonstration 6	257
Use of throws Keyword.....	260
Demonstration 7	261
Demonstration 8	263

TABLE OF CONTENTS

- Checked Versus Unchecked Exceptions..... 265
 - Q&A Session 266
- Discussion on Chained Exceptions 266
 - Demonstration 9 267
- Creating a Custom Exception..... 272
 - Demonstration 10 273
 - Q&A Session 276
 - Demonstration 11 276
 - Q&A Session 277
 - Demonstration 12 278
- Summary..... 280
- Chapter 11: Thread Programming 281**
 - Creating Threads..... 283
 - Extending the Thread Class..... 284
 - Demonstration 1 285
 - Implementing the Runnable Interface..... 288
 - Demonstration 2 288
 - Q&A Session 291
 - Demonstration 3 292
 - Case Study with Different Thread Class Methods 300
 - Demonstration 4 301
 - Q&A Session 303
 - Synchronization 305
 - Use of Synchronized Methods..... 306
 - Demonstration 5 307
 - Use of Synchronized Block 309
 - Demonstration 6 309
 - Deadlock 311
 - Types of Deadlock 311
 - Demonstration 7 312
 - Detecting Deadlocks in the System..... 314

Interthread Communication	317
Demonstration 8	319
Lifecycle of a Thread	323
Q&A Session	325
Summary.....	327
Chapter 12: Generic Programming	329
Compare Generic Programs with Non-Generic Programs.....	329
Demonstration 1	329
Demonstration 2	331
Demonstration 3	335
Demonstration 4	337
Wildcard Types in Generic Programming	340
Upper-bound Wildcard	340
Demonstration 5	342
Lower-bound Wildcard.....	346
Demonstration 6	346
Unbounded Wildcard	349
Q&A Session	350
Demonstration 6A.....	354
Bounded Type Parameter	355
Demonstration 7	357
Demonstration 8	360
Q&A Session	361
Erasures	361
Demonstration 9	362
Raw Types	366
Demonstration 10	367
Q&A Session	368

TABLE OF CONTENTS

- Type Inference Using Diamond Operator..... 369
- Applying Inheritance 370
 - Demonstration 11 370
 - Bridge Method 372
 - Demonstration 12..... 373
- Important Restrictions in Generic Programming..... 375
 - Don't Instantiate Generic Types with Primitive Types 376
 - Your Generic Class Cannot Subclass Directly or Indirectly from Throwable..... 376
 - You Cannot Overload a Method Where the Formal Parameter Types of Each Overload Are Erased to the Same Raw Type..... 376
 - Static Field Type Parameter Is Not Allowed in Your Generic Class 377
 - You Cannot Instantiate the Type Parameters In Your Generic Class 377
 - One Final Suggestion..... 378
- Summary..... 378
- Chapter 13: Database Programming 381**
 - Database and DBMS 382
 - Types of DBMS..... 382
 - RDBMS 383
 - SQL 384
 - Connecting to a Database 385
 - Q&A Session 387
 - Talking to a Database in a Java Application..... 388
 - Important Terms..... 390
 - Creating a Database and Inserting Records..... 392
 - MySQL Command Prompt View 395
 - Demonstration 1 397
 - Demonstration 2 399
 - Demonstration 3 403
 - Q&A Session 408
 - Demonstration 4 410

Q&A Session	417
Demonstration 5	418
Summary.....	421
Chapter 14: Important Features in Java’s Enhancement Path.....	423
Try-with-resource from Java 7	423
Demonstration 1	424
Q&A Session	425
Implementing Functional Interface Methods Using Lambda Expressions from Java 8	425
Demonstration 2	426
Q&A Session	427
Private Interface Method from Java 9.....	428
Demonstration 3	428
Q&A Session	430
Local Variable Type Inference from Java 10.....	432
Demonstration 4	432
Restrictions	433
New String Methods from Java 11	434
Demonstration 5	434
Q&A Session	435
New switch Expression in Java 12/13	437
Demonstration 6	437
Q&A Session	439
Summary.....	439
Part III: Explore Real-World Scenarios	441
Chapter 15: Introduction to Design Patterns	443
Key Points	444
Creational Patterns.....	445
Structural Patterns	446

TABLE OF CONTENTS

- Behavioral Patterns 446
- Class and Object Patterns 447
- Q&A Session 447
- Prototype Pattern 449
 - GoF Definition 449
 - Concept 449
 - Real-life Example 450
 - Coding Example 450
 - Illustration..... 451
 - Class Diagram 452
 - Package Explorer View 453
 - Implementation 453
 - Q&A Session 456
 - Shallow Copy Versus Deep Copy in Java..... 459
- Bridge Pattern 465
 - GoF Definition 465
 - Concept 465
 - Real-life Example 465
 - Coding Example 466
 - Illustration..... 466
 - Class Diagram 469
 - Package Explorer View 469
 - Implementation 470
 - Q&A Session 475
- Observer Pattern 477
 - GoF Definition 477
 - Concept 477
 - Real-life Example 481
 - Coding Example 481
 - Illustration..... 482

Class Diagram	482
Package Explorer View	483
Implementation	484
Q&A Session	488
Summary.....	492
Chapter 16: Frequently Asked Questions.....	493
Appendix A: Test Your Skill in Language Fundamentals.....	501
SET 1	501
SET 2.....	503
SET 3.....	507
SET 4.....	509
SET 5.....	512
SET 6.....	513
SET 7.....	521
SET 8.....	528
SET 9.....	534
Appendix B: Getting Started with Java.....	537
Basic Terms.....	538
Installation	540
Download JDK	540
An Important License Update from Oracle.....	541
Download Eclipse	542
Naming Conventions.....	542
Appendix C: Installing MySQL and Testing SQL Commands.....	545
Index.....	581

About the Author



Vaskaran Sarcar obtained his Master of Engineering degree from Jadavpur University, Kolkata, India. He was a national Gate Scholar and has more than twelve years of experience in education and the IT industry. He worked as senior software engineer, specialist, and team lead in the R&D Hub at HP Inc. India until August 2019. He is an alumnus of prestigious institutions in India, such as Jadavpur University, Vidyasagar University, and Presidency University (formerly Presidency College). He loves to spend time with his kids and family members. Reading and learning new things are his passions. Other books by Vaskaran include the following:

- *Java Design Patterns*, Second Edition (Apress, 2018)
- *Design Patterns in C#* (Apress, 2018)
- *Interactive C#* (Apress, 2017)
- *Interactive Object-Oriented Programming in Java* (Apress, 2016)
- *Java Design Patterns*, First Edition (Apress, 2016)
- *C# Basics: Test Your Skill* (CreateSpace, 2015)
- *Operating System: Computer Science Interview Series* (CreateSpace, 2014)

About the Technical Reviewer



Yogesh Sharma is a full-stack engineer at Mphasis Pvt Ltd where he primarily focuses on modernising existing stacks with the help of containerisation, microservices and prevalent DevOps disciplines. With his recently discovered love for Infrastructure as Code and Adaptive Intelligence, he enjoys travelling, voluntary services and watching sitcoms. He would also like to take this opportunity to thank his wife, Akanksha for supporting him in all his endeavors.

Acknowledgments

At first, I thank the Almighty. I sincerely believe that with *his* blessings only, I completed this book. I also extend my deepest gratitude and thanks to the following:

Ratanlal Sarkar and **Manikuntala Sarkar**: My dear parents, with your blessings only, could I complete this work.

Indrani, my wife, and **Ambika**, my daughter: Sweethearts, once again, without your love I could not proceed at all. I know that we needed to limit many social gatherings and invitations so I could complete this work on time, and each time I promised you that I would take a long break and spend more time with you.

Sambaran, my brother: Thank you for your constant encouragement toward me.

Yogesh: You are technical advisor for this second edition. I know that whenever I was in need, your support was there. Thank you one more time.

Shekhar and **Anupam**: My friends and technical advisors in the previous edition of this book. Though this time you were not involved, I still acknowledge your support and help in the development of *Interactive Object-Oriented Programming in Java*, First Edition.

Avirup Mullick: I first met you in 1998. You were my college senior. A special thanks to you for taking time to write a foreword for my book. From the moment experts like you agreed to write for me, I got additional motivation to enhance the quality of my work.

Celestin: Thanks for giving me another opportunity to work with you and Apress.

April, Nirmal, Arockia Rajan, Ramraj, Chinnaparaj, and **Vinoth**: Thank you for your exceptional support to beautify my work.

Lastly, I extend my deepest gratitude to my publisher, the editorial board members, and everyone who directly or indirectly supported this book.

Preface

It is my privilege to present you with *Interactive Object-Oriented Programming in Java: Learn and Test Your Programming Skills* (Second Edition).

If you are curious about what the most important and unique characteristics of this book are, I would say that it is interactive and very simple. The goal was not to demonstrate typical and tough programs using all the latest features of Java. On the contrary, the true goal is to fuel your creativity by using the core constructs in Java. The word *core* is more important than *the latest* when you learn a new technology. Whatever is the latest today will be outdated tomorrow. But core concepts are evergreen.

This book focuses on the implementation of object-oriented programming concepts using the most basic features of Java so that you don't need to be familiar with advanced Java topics. The examples are simple and straightforward. I believe that these examples are written in such a way that even if you are familiar with another popular language, such as C#, C++, and so on, you can still easily grasp the concepts in this book.

You'll probably agree that when you travel an unknown path to a destination, it can help if you get a loving and caring guide. Learning a new programming language through a book is also a journey, a fact that was always on my mind as I wrote. So, in this book, I did not explain a topic in only an informative way. Instead, I made this book interactive, with one or more "Q&A Sessions" in each chapter. These sessions will not only assist you in your learning process, but can also act as a "doubt-clearing session" because they will feel like you are asking your guide some questions (or expressing your doubts) and that you are receiving the answers from him in a simple one-to-one communication. In addition to this, in most of the cases, you'll get a full demonstration of a program with output analysis so that you can get the maximum benefit.

In short, the aim of this book is to help you to get a feel of a Java classroom environment. I have been involved in teaching since 2005. I have taken classes in both engineering and non-engineering colleges. And, fortunately, most of my teaching involvement was based on Java and its advanced topics. That is the true motivation for why I wanted to introduce a book like this. Before you jump into the topics, let me highlight a few points about the book, its chapters' organization, and the intended readers.

PREFACE

The book has three major parts. The first nine chapters make up Part 1, in which you will see the discussion and implementation of object-oriented concepts in Java. Part 2 consists of five more chapters (from Chapter 10 to Chapter 14). In Part 2, you will explore something from “Advanced Java,” where you will learn about exception handling, multi-thread programming, generic programming, and JDBC programming. In Chapter 14, you will get to know about the feature evolution path, where you will experiment with important features that come in different versions of Java. But I have picked only those features that enhance what you will learn in Part 1 of the book, so that you can understand how these upgraded features can make your programming life easier. Finally, in Part 3 of the book, you will learn about some real-world implementation using three important design patterns. Part 3 also consists of a chapter of FAQs, which is basically a subset of all the Q&A Sessions in this book. It can provide a quick review of all the topics that you learn in this book.

The target readers for this book are those who know the basic language constructs in Java and how to compile or run a simple Java application. This book does not invest time in topics that are easily available online, such as how to install Eclipse on your system, or how to write a “Hello World” program in Java, or how can you use an `if-else` statement or a `while` loop in your Java program and so forth. Instead, the book starts with a discussion in object-oriented programming. So, I expect that before you enter into Chapter 1, you will be familiar with simple Java programs and your coding environment is ready. My discussion with you starts with the object-oriented concepts you can use in Java. Here, I focus on the fundamental features of Java, and I also explain how these concepts can be learned and used effectively.

But do not worry! To assist you with asking/thinking better questions in doubt-clearing sessions, an entire section is added at the end of the book (Appendix A). This appendix discusses some key concepts in Java and helps you evaluate your skills in the language basics. You may need to come back to this section many times because it acts as a reference. Even if you do not know all of these topics, gradually, upon repeated practice, you will become familiar with them. So, if you are new to programming or if you have some idea about other programming languages, this section can assist you a lot. It can also help you prepare for a job interview or an examination by answering some tricky questions that may seem very easy at the beginning.

I said earlier that in this book each chapter contains one or more Q&A Sessions, which will give you a feel of learning in a classroom environment—where your teacher will discuss some problems or topics, ask you questions, and allow you to ask counter

questions. If you are dedicated to this subject and think deeply about the questions and the corresponding answers, you will surely develop confidence in this language.

In a semester, you need to attend a certain number of lectures to complete the fundamental topics, and you know that learning is a continuous process. So, this book is not for those who want to learn Java in 24 hours or in 7 days. It is up to you only. I can only say that the book is designed for you in such a way that upon its completion, you will have developed an adequate knowledge of the topic, you will have learned the key features of this powerful language and object-oriented programming, and you will have learned how you should write programs in Java and, most important, how to go further.

I have taken care to provide codes that are compatible with all the latest versions of Java. Also, it is not mandatory for you to learn Eclipse. You can simply run these programs in your preferred IDE (integrated development environment). I have chosen Eclipse because it is widely used to develop Java applications.

Please remember that as you learn about these concepts, try writing your own code; only then will you master this area. You can always share your comments to truly complete this book and enhance your future work.

You will be able to download all the source code of the book from the publisher's website. I have a plan to maintain the "Errata" and, if required, I can also make some updates/announcements there. So, it is suggested that you visit those pages to receive the corrections or updates, if any.

Lastly, I hope that this enhanced edition can provide more help to you and that you will like the book.

Who Is This Book For?

In short, **you can pick** this book if the answer is “yes” to the following questions:

- Are you familiar with basic constructs in Java?
- Do you know how to set up your coding environment?
- Do you want to explore object-oriented programming step-by-step?
- Do you want to review your understanding of basic programming skills in Java?
- Do you want to explore something from advanced Java (for example, generic programming, JDBC Programming, multi-thread programming, etc.) ?
- Are you interested to know about some real-world implementation techniques?

Probably you **shouldn't pick** this book if the answer is “yes” to any of the following questions:

- Are you totally new to Java?
- Are you looking for all the advanced concepts in Java in depth?
- Are you interested in exploring only the latest features of Java?
- Do you dislike a book that has an emphasis on Q&A sessions?
- “I do not like Windows and Eclipse. I want to learn Java without them.” Is this statement true for you?

Guidelines for Using This Book

Here are some suggestions so you can use this book more effectively:

- If you are confident with the topics covered in Appendix A, you can start with Chapter 1 of the book. I suggest you go through the chapters sequentially. Some fundamental questions may be discussed in the Q&A Session of a previous chapter, and I have not repeated those in the later chapters.
- These programs are tested with Java 8 (update 172), and I have used Eclipse IDE in a Windows 10 environment. When I started the second edition of the book, Photon was the latest edition of Eclipse (released June 27, 2018), Java 8 was the long-term support (LTS) version, and Java 10 was the rapid-release version. Java 11 is the next LTS version after Java 8 and was planned for September 2018. But as it turned out, by the time I finished my work, Java 13 and Eclipse Java 2019-09 had been released. But all these versions' details should not matter to you, because I have used the most basic constructs of Java. So, I believe that these codes should execute smoothly in the upcoming versions of Java/Eclipse as well. To experiment with this, I tested some portions of these codes in different systems and different environments (including online editors), and I always received the expected output. With these experiments, I believe that the results should not vary in other environments as well, but you know the nature of software—they can misbehave and surprise you. So, I recommend that if you want to see the exact same outputs that are shown in the book, it will be better if you can mimic the same environment.
- There is an exception for Chapter 14 codes. Some of them use the latest Java features, and my Eclipse environment was not ready to accommodate the changes. So, I executed some programs with the latest features in a command-line environment. You can do the same.

GUIDELINES FOR USING THIS BOOK

- In some examples, to draw class diagrams, ObjectAid Uml Explorer is used in the Eclipse editor. It is a lightweight tool for Eclipse. At the time of this writing, it is free if you want to draw the class diagrams, but to draw the sequence diagrams, you will need to purchase a license. The online link <http://www.objectaid.com/home> can give you the details of these licenses and terms and conditions.

Conventions Used in This Book

All programs in this book are organized under package statements like the following:

```
package java2e.chapter2;
```

```
class ClassEx1 {  
    // Field initialization is optional.  
    // Here myInt is initialized with the value 25.  
    public int myInt = 25;  
    // In the following case, it will be initialized with default //value 0.  
}  
  
public class Demonstration1 {  
  
    public static void main(String[] args) {  
        System.out.println("***Demonstration-1. A class demo with 2  
        objects ***");  
        ClassEx1 obA = new ClassEx1();  
        ClassEx1 obB = new ClassEx1();  
        System.out.println("obA.myInt = " + obA.myInt);  
        System.out.println("obB.myInt = " + obB.myInt);  
    }  
}
```

It simply says that there is folder classed java2e, inside which is another folder chapter2, and you are storing Demonstration1.java inside it. So, the Demonstration1.class and ClassEx1.class files should be available there. You'll learn about Java packages in Chapter 7. Once you're familiar with them, you'll acknowledge that it is always a better practice to organize your codes with packages.

But when you start learning the concepts, to compile and run the programs, *the package statements are NOT mandatory for you*. So, initially you can play with all these programs without the package statements. And, you may store your programs in your preferred location, so, at your end, you may need to modify the package statements when you use them.

CONVENTIONS USED IN THIS BOOK

All the outputs and codes of the book follow the same font and structure. To draw your attention, in some places, I have made them bold like the following:

Demonstration-17.A comparison study:Using a final class vs using a private constructor

Called the private constructor.

Setting the default the value x=10.

Exit-private non-parameterized constructor.

Updating the default value of x.

Exit-parameterised constructor.

The parent.x=15

Called the private constructor.

Setting the default the value x=10.

Exit-private non-parameterized constructor.

Updating the default value of x.

Exit-parameterised constructor.

The child.x=2

The child.y=3

Foreword

It is an absolute honor and privilege to write this foreword for Vaskaran's latest book, *Interactive Object-Oriented Programming in Java: Learn and Test Your Programming Skills*.

I have known Vaskaran for many years now. From studying in the same institutions, such as Presidency University (formerly Presidency College) and Vidyasagar University, to working in the IT industry, we tread our own paths in our academic and professional careers, but he always found a way to give back to the community. This is a unique trait, and I admire him for that.

His passion for writing and his desire to share the knowledge he has acquired through his years of experience in the industry are reflected in this book, where the approach to learning is through clarifying theoretical concepts while testing one's programming skills. This practical approach will be invaluable to readers and will help them in real-life programming situations in their exams, interviews, or jobs.

This book discusses the fundamental concepts of object-oriented programming in depth, with examples in Java. It also covers some advance concepts, including design patterns in Java. It is an easy read, and each concept has been handled with precision: fundamentals have been clarified for in-depth knowledge building while demonstrations and examples have kept it interesting. Each chapter is augmented with a Q & A Session, which will be helpful in putting things in perspective or clearing one's doubts by understanding the pros and cons of each of these patterns. The "Test Your Skill in Language Basics" section will help readers review their understanding of the fundamentals of Java before advancing their expertise in the field.

The FAQ chapter at the end of the book is particularly helpful in refreshing one's knowledge and getting one in the mindset to crack those technical interviews with confidence.

This book is a must-read for Java developers or aspiring Java developers, as it will most certainly enhance one's object-oriented programming skills in Java to a great extent. For that matter, even developers from a non-Java background can benefit from the book.

I wish Vaskaran and this book all the success it deserves.

Avirup Mullick
Manager, Global Operations Center, Adobe Systems

PART I

Fundamentals of Object-Oriented Programming

Chapter 1: Object-Oriented Programming Concepts

Chapter 2: The Building Blocks: Class and Objects

Chapter 3: Class and Objects in Depth

Chapter 4: The Concept of Inheritance

Chapter 5: Get Familiar with Polymorphism

Chapter 6: Abstract Class and Interface: The True Art in OOP

Chapter 7: Package

Chapter 8: Understanding Class Variables and Class Methods

Chapter 9: Quick Recap of OOP Principles