



Integrated Cloud Applications & Platform Services



Java SE 8 Programming

Student Guide - Volume I

D84838GC10

Edition 1.0 | December 2016 | D87757

Learn more from Oracle University at education.oracle.com

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font with a registered trademark symbol, set against a red background.

Authors

Anjana Shenoy
Michael Williams
Tom McGinn
Peter Fernandez

Technical Contributors and Reviewers

Pete Daly
Sravanti Tatiraju
Nick Ristuccia
Stuart Marks
Hiroshi Hiraga
Peter Hall
Matthew Slingsby
Marcus Hirt
Irene Rusman
Joanne Sun
Marilyn Beck
Joe A Boulenuar

Editors

Aju Kumar
Malavika Jinka
Arijit Ghosh
Anwesha Ray

Graphic Designer

Divya Thallap

Publishers

Giri Venugopal
Michael Sebastian
Veena Narasimhan

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

1 Introduction

Course Goals	1-2
Course Objectives	1-3
Audience	1-5
Prerequisites	1-6
Class Introductions	1-7
Course Environment	1-8
Java Programs Are Platform-Independent	1-9
Java Technology Product Groups	1-10
Java SE Platform Versions	1-11
Downloading and Installing the JDK	1-12
Java in Server Environments	1-13
The Internet of Things	1-14
The Java Community	1-15
The Java Community Process (JCP)	1-16
OpenJDK	1-17
Oracle Java SE Support	1-18
Additional Resources	1-19
Summary	1-20

2 Java Syntax and Class Review

Objectives	2-2
Java Language Review	2-3
Java Class Structure	2-4
A Simple Class	2-5
Java Naming Conventions	2-6
How to Compile and Run	2-7
How to Compile and Run: Example	2-8
Code Blocks	2-9
Primitive Data Types	2-10
Numeric Literals	2-11
Operators	2-12
Logical Operators	2-13
if else Statement	2-14
switch Statement	2-15

- while Loop 2-16
- do-while Loop 2-17
- for Loop 2-18
- Arrays and for-each Loop 2-19
- Strings 2-20
- String Operations: StringBuilder 2-21
- A Simple Java Class: Employee 2-22
- Methods 2-23
- Creating an Instance of a Class 2-24
- Constructors 2-25
- package Statement 2-26
- import Statements 2-27
- Java Is Pass-By-Value 2-29
- Pass-By-Value for Object References 2-30
- Objects Passed as Parameters 2-31
- Garbage Collection 2-32
- Summary 2-33
- Practice 2-1 Overview: Creating Java Classes 2-34
- Quiz 2-35

3 Encapsulation and Subclassing

- Objectives 3-2
- Encapsulation 3-3
- Encapsulation: Example 3-4
- Encapsulation: Public and Private Access Modifiers 3-5
- Encapsulation: Private Data, Public Methods 3-6
- Employee Class Refined 3-7
- Make Classes as Immutable as Possible 3-8
- Method Naming: Best Practices 3-9
- Encapsulation: Benefits 3-10
- Creating Subclasses 3-11
- Subclassing 3-12
- Manager Subclass 3-13
- Constructors in Subclasses 3-14
- Using super 3-15
- Constructing a Manager Object 3-16
- Overloading Methods 3-17
- Overloaded Constructors 3-18
- Overloaded Constructors: Example 3-19
- Single Inheritance 3-20
- Summary 3-21

Practice 3-1 Overview: Creating Subclasses 3-22

Quiz 3-23

4 Overriding Methods, Polymorphism, and Static Classes

Objectives 4-2

Using Access Control 4-3

Protected Access Control: Example 4-4

Access Control: Good Practice 4-5

Overriding Methods 4-6

Invoking an Overridden Method 4-8

Virtual Method Invocation 4-9

Accessibility of Overriding Methods 4-10

Applying Polymorphism 4-11

Using the instanceof Keyword 4-13

Overriding Object methods 4-14

Object toString Method 4-15

Object equals Method 4-16

Overriding equals in Employee 4-17

Overriding Object hashCode 4-18

Methods Using Variable Arguments 4-19

Casting Object References 4-21

Upward Casting Rules 4-22

Downward Casting Rules 4-23

static Keyword 4-24

Static Methods 4-25

Using Static Variables and Methods: Example 4-26

Implementing Static Methods 4-27

Calling Static Methods 4-28

Static Variables 4-29

Defining Static Variables 4-30

Using Static Variables 4-31

Static Initializers 4-32

Static Imports 4-33

Design Patterns 4-34

Singleton Pattern 4-35

Singleton: Example 4-36

Immutable Classes 4-37

Example: Creating Immutable class in Java 4-38

Summary 4-39

Practice 4-1 Overview: Overriding Methods and Applying Polymorphism 4-40

Practice 4-2 Overview: Overriding Methods and Applying Polymorphism 4-41

Practice 4-3 Overview: Applying the Singleton Design Pattern 4-42
Quiz 4-43

5 Abstract and Nested Classes

Objectives 5-2
Modeling Business Problems with Classes 5-3
Enabling Generalization 5-4
Identifying the Need for Abstract Classes 5-5
Defining Abstract Classes 5-6
Defining Abstract Methods 5-7
Validating Abstract Classes 5-8
Final Methods 5-9
Final Classes 5-10
Final Variables 5-11
Declaring Final Variables 5-12
Nested Classes 5-13
Example: Member Class 5-14
Enumerations 5-15
Enum Usage 5-16
Complex Enums 5-17
Summary 5-19
Practice 5-1 Overview: Applying the Abstract Keyword 5-20
Practice 5-2 Overview: Using Inner Class As a Helper Class 5-21
Practice 5-3 Overview: Using Java Enumerations 5-22
Quiz 5-23

6 Interfaces and Lambda Expressions

Objectives 6-2
Java Interfaces 6-3
A Problem Solved by Interfaces 6-4
CrushedRock Class 6-5
The SalesCalcs Interface 6-6
Adding an Interface 6-7
Interface References 6-8
Interface Reference Usefulness 6-9
Interface Code Flexibility 6-10
default Methods in Interfaces 6-11
default Method: Example 6-12
static Methods in Interfaces 6-13
Constant Fields 6-14
Extending Interfaces 6-15

- Implementing and Extending 6-16
- Anonymous Inner Classes 6-17
- Anonymous Inner Class: Example 6-18
- String Analysis Regular Class 6-19
- String Analysis Regular Test Class 6-20
- String Analysis Interface: Example 6-21
- String Analyzer Interface Test Class 6-22
- Encapsulate the for Loop 6-23
- String Analysis Test Class with Helper Method 6-24
- String Analysis Anonymous Inner Class 6-25
- String Analysis Lambda Expression 6-26
- Lambda Expression Defined 6-27
- What Is a Lambda Expression? 6-28
- Lambda Expression Shorthand 6-31
- Lambda Expressions as Variables 6-32
- Summary 6-33
- Practice 6-1: Implementing an Interface 6-34
- Practice 6-2: Using Java Interfaces 6-35
- Practice 6-3: Creating Lambda Expression 6-36
- Quiz 6-37

7 Generics and Collections

- Objectives 7-2
- Topics 7-3
- Generics 7-4
- Simple Cache Class Without Generics 7-5
- Generic Cache Class 7-6
- Generics in Action 7-7
- Generics with Type Inference Diamond 7-8
- Collections 7-9
- Collection Types 7-10
- Collection Interfaces and Implementation 7-11
- List Interface 7-12
- ArrayList 7-13
- Autoboxing and Unboxing 7-14
- ArrayList Without Generics 7-15
- Generic ArrayList 7-16
- Generic ArrayList: Iteration and Boxing 7-17
- Set Interface 7-18
- TreeSet: Implementation of Set 7-19
- Map Interface 7-20

Map Types 7-21
TreeMap: Implementation of Map 7-22
Deque Interface 7-23
Stack with Deque: Example 7-24
Ordering Collections 7-25
Comparable: Example 7-26
Comparable Test: Example 7-27
Comparator Interface 7-28
Comparator: Example 7-29
Comparator Test: Example 7-30
Summary 7-31
Practice 7-1 Overview: Counting Part Numbers by Using a HashMap 7-32
Practice 7-2 Overview: Implementing Stack by Using a Deque Object 7-33
Quiz 7-34

8 Collections, Streams, and Filters

Objectives 8-2
Collections, Streams, and Filters 8-3
The Person Class 8-4
Person Properties 8-5
Builder Pattern 8-6
Collection Iteration and Lambdas 8-7
RoboCallTest07: Stream and Filter 8-8
RobocallTest08: Stream and Filter Again 8-9
SalesTxn Class 8-10
Java Streams 8-11
The Filter Method 8-12
Method References 8-13
Method Chaining 8-14
Pipeline Defined 8-16
Summary 8-17
Practice Overview 8-18

9 Lambda Built-in Functional Interfaces

Objectives 9-2
Built-in Functional Interfaces 9-3
The java.util.function Package 9-4
Example Assumptions 9-5
Predicate 9-6
Predicate: Example 9-7
Consumer 9-8

Consumer: Example 9-9
Function 9-10
Function: Example 9-11
Supplier 9-12
Supplier: Example 9-13
Primitive Interface 9-14
Return a Primitive Type 9-15
Return a Primitive Type: Example 9-16
Process a Primitive Type 9-17
Process Primitive Type: Example 9-18
Binary Types 9-19
Binary Type: Example 9-20
Unary Operator 9-21
UnaryOperator: Example 9-22
Wildcard Generics Review 9-23
Summary 9-24
Practice Overview 9-25

10 Lambda Operations

Objectives 10-2
Streams API 10-3
Types of Operations 10-4
Extracting Data with Map 10-5
Taking a Peek 10-6
Search Methods: Overview 10-7
Search Methods 10-8
Optional Class 10-9
Lazy Operations 10-10
Stream Data Methods 10-11
Performing Calculations 10-12
Sorting 10-13
Comparator Updates 10-14
Saving Data from a Stream 10-15
Collectors Class 10-16
Quick Streams with Stream.of 10-17
Flatten Data with flatMap 10-18
Summary 10-19
Practice Overview 10-20

11 Exceptions and Assertions

- Objectives 11-2
- Error Handling 11-3
- Exception Handling in Java 11-4
- try-catch Statement 11-5
- Exception Objects 11-6
- Exception Categories 11-7
- Handling Exceptions 11-8
- finally Clause 11-9
- try-with-resources Statement 11-10
- Catching Multiple Exceptions 11-11
- Declaring Exceptions 11-12
- Handling Declared Exceptions 11-13
- Throwing Exceptions 11-14
- Custom Exceptions 11-15
- Assertions 11-16
- Assertion Syntax 11-17
- Internal Invariants 11-18
- Control Flow Invariants 11-19
- Class Invariants 11-20
- Controlling Runtime Evaluation of Assertions 11-21
- Summary 11-22
- Practice 11-1 Overview: Catching Exceptions 11-23
- Practice 11-2 Overview: Extending Exception and Using throw and throws 11-24
- Quiz 11-25

12 Java Date/Time API

- Objectives 12-2
- Why Is Date and Time Important? 12-3
- Previous Java Date and Time 12-4
- Java Date and Time API: Goals 12-5
- Working with Local Date and Time 12-6
- Working with LocalDate 12-7
- LocalDate: Example 12-8
- Working with LocalTime 12-9
- LocalTime: Example 12-10
- Working with LocalDateTime 12-11
- LocalTimeDate: Example 12-12
- Working with Time Zones 12-13
- Daylight Savings Time Rules 12-14
- Modeling Time Zones 12-15

Creating ZonedDateTime Objects 12-16
Working with ZonedDateTime Gaps/Overlaps 12-17
ZoneRules 12-18
Working Across Time Zones 12-19
Date and Time Methods 12-20
Date and Time Amounts 12-21
Period 12-22
Duration 12-23
Calculating Between Days 12-24
Making Dates Pretty 12-25
Using Fluent Notation 12-26
Summary 12-27
Practices 12-28

13 Java I/O Fundamentals

Objectives 13-2
Java I/O Basics 13-3
I/O Streams 13-4
I/O Application 13-5
Data Within Streams 13-6
Byte Stream InputStream Methods 13-7
Byte Stream OutputStream Methods 13-8
Byte Stream: Example 13-9
Character Stream Reader Methods 13-10
Character Stream Writer Methods 13-11
Character Stream: Example 13-12
I/O Stream Chaining 13-13
Chained Streams: Example 13-14
Console I/O 13-15
Writing to Standard Output 13-16
Reading from Standard Input 13-17
Channel I/O 13-18
Persistence 13-19
Serialization and Object Graphs 13-20
Transient Fields and Objects 13-21
Transient: Example 13-22
Serial Version UID 13-23
Serialization: Example 13-24
Writing and Reading an Object Stream 13-25
Serialization Methods 13-26
readObject: Example 13-27

Summary 13-28
Practice 13-1 Overview: Writing a Simple Console I/O Application 13-29
Practice 13-2 Overview: Serializing and Deserializing a ShoppingCart 13-30
Quiz 13-31

14 Java File I/O (NIO.2)

Objectives 14-2
New File I/O API (NIO.2) 14-3
Limitations of java.io.File 14-4
File Systems, Paths, Files 14-5
Relative Path Versus Absolute Path 14-6
Java NIO.2 Concepts 14-7
Path Interface 14-8
Path Interface Features 14-9
Path: Example 14-10
Removing Redundancies from a Path 14-11
Creating a Subpath 14-12
Joining Two Paths 14-13
Symbolic Links 14-14
Working with Links 14-15
File Operations 14-16
Checking a File or Directory 14-17
Creating Files and Directories 14-19
Deleting a File or Directory 14-20
Copying a File or Directory 14-21
Moving a File or Directory 14-22
List the Contents of a Directory 14-23
Walk the Directory Structure 14-24
BufferedReader File Stream 14-25
NIO File Stream 14-26
Read File into ArrayList 14-27
Managing Metadata 14-28
Symbolic Links 14-29
Summary 14-30
Practice Overview 14-31
Quiz 14-33

15 Concurrency

Objectives 15-2
Task Scheduling 15-3
Legacy Thread and Runnable 15-4

Extending Thread	15-5
Implementing Runnable	15-6
The java.util.concurrent Package	15-7
Recommended Threading Classes	15-8
java.util.concurrent.ExecutorService	15-9
Example ExecutorService	15-10
Shutting Down an ExecutorService	15-11
java.util.concurrent.Callable	15-12
Example Callable Task	15-13
java.util.concurrent.Future	15-14
Example	15-15
Threading Concerns	15-16
Shared Data	15-17
Problems with Shared Data	15-18
Nonshared Data	15-19
Atomic Operations	15-20
Out-of-Order Execution	15-21
The synchronized Keyword	15-22
synchronized Methods	15-23
synchronized Blocks	15-24
Object Monitor Locking	15-25
Threading Performance	15-26
Performance Issue: Examples	15-27
java.util.concurrent Classes and Packages	15-28
The java.util.concurrent.atomic Package	15-29
java.util.concurrent.CyclicBarrier	15-30
Thread-Safe Collections	15-32
CopyOnWriteArrayList: Example	15-33
Summary	15-34
Practice 15-1 Overview: Using the java.util.concurrent Package	15-35
Quiz	15-36

16 The Fork-Join Framework

Objectives	16-2
Parallelism	16-3
Without Parallelism	16-4
Naive Parallelism	16-5
The Need for the Fork-Join Framework	16-6
Work-Stealing	16-7
A Single-Threaded Example	16-8
java.util.concurrent.ForkJoinTask<V>	16-9

- RecursiveTask Example 16-10
- compute Structure 16-11
- compute Example (Below Threshold) 16-12
- compute Example (Above Threshold) 16-13
- ForkJoinPool Example 16-14
- Fork-Join Framework Recommendations 16-15
- Summary 16-16
- Practice 16-1 Overview: Using the Fork-Join Framework 16-17
- Quiz 16-18

17 Parallel Streams

- Objectives 17-2
- Streams Review 17-3
- Old Style Collection Processing 17-4
- New Style Collection Processing 17-5
- Stream Pipeline: Another Look 17-6
- Styles Compared 17-7
- Parallel Stream 17-8
- Using Parallel Streams: Collection 17-9
- Using Parallel Streams: From a Stream 17-10
- Pipelines Fine Print 17-11
- Embrace Statelessness 17-12
- Avoid Statefulness 17-13
- Streams Are Deterministic for Most Part 17-14
- Some Are Not Deterministic 17-15
- Reduction 17-16
- Reduction Fine Print 17-17
- Reduction: Example 17-18
- A Look Under the Hood 17-24
- Illustrating Parallel Execution 17-25
- Performance 17-36
- A Simple Performance Model 17-37
- Summary 17-38
- Practice 17-39

18 Building Database Applications with JDBC

- Objectives 18-2
- Using the JDBC API 18-3
- Using a Vendor's Driver Class 18-4
- Key JDBC API Components 18-5
- Writing Queries and Getting Results 18-6

- Using a ResultSet Object 18-7
- CRUD Operations Using JDBC API: Retrieve 18-8
- CRUD Operations Using JDBC: Retrieve 18-9
- CRUD Operations Using JDBC API: Create 18-10
- CRUD Operations Using JDBC API: Update 18-11
- CRUD Operations Using JDBC API: Delete 18-12
- SQLException Class 18-13
- Closing JDBC Objects 18-14
- try-with-resources Construct 18-15
- Using PreparedStatement 18-16
- Using PreparedStatement: Setting Parameters 18-17
- Executing PreparedStatement 18-18
- PreparedStatement: Using a Loop to Set Values 18-19
- Using CallableStatement 18-20
- Summary 18-21
- Practice 18-1 Overview: Working with the Derby Database and JDBC 18-22
- Quiz 18-23

19 Localization

- Objectives 19-2
- Why Localize? 19-3
- A Sample Application 19-4
- Locale 19-5
- Properties 19-6
- Loading and Using a Properties File 19-7
- Loading Properties from the Command Line 19-8
- Resource Bundle 19-9
- Resource Bundle File 19-10
- Sample Resource Bundle Files 19-11
- Initializing the Sample Application 19-12
- Sample Application: Main Loop 19-13
- The printMenu Method 19-14
- Changing the Locale 19-15
- Sample Interface with French 19-16
- Format Date and Currency 19-17
- Displaying Currency 19-18
- Formatting Currency with NumberFormat 19-19
- Displaying Dates 19-20
- Displaying Dates with DateTimeFormatter 19-21
- Format Styles 19-22
- Summary 19-23

Practice 19-1 Overview: Creating a Localized Date Application 19-24
Quiz 19-25

20 Oracle Cloud

Agenda 20-2
What is Cloud? 20-3
What is Cloud Computing? 20-4
History – Cloud Evolution 20-5
Components of Cloud Computing 20-6
Characteristics of Cloud 20-7
Cloud Deployment Models 20-8
Cloud Service Models 20-9
Industry Shifting from On-Premises to the Cloud 20-13
Oracle IaaS Overview 20-15
Oracle PaaS Overview 20-16
Oracle SaaS Overview 20-17
Summary 20-18

21 Oracle Application Container Cloud Service Overview

Objectives 21-2
Oracle Application Container Cloud Service 21-3
Oracle Application Container Cloud 21-4
Polyglot Platform 21-5
Open Platform 21-6
Container-based Application Platform as a Service 21-7
Elastic Scaling 21-8
Profiling 21-9
Manageable 21-10
Deploy—Application Archive (Zip) 21-12
Application Deployment 21-13
Application Container Cloud Architecture 21-14
Load Balancer 21-15
Oracle Developer Cloud Service 21-16
Developer Cloud Service – Easy Adoption/Integration 21-17
Application Container Cloud Service Advantages 21-19
Summary 21-20