

# Table of Contents

1. [Cover](#)
2. [Acknowledgments](#)
3. [About the Authors](#)
4. [Introduction](#)
  1. [Understanding the Exam](#)
  2. [Reading This Book](#)
  3. [Preparing for the Exam](#)
  4. [Taking the Exam](#)
  5. [Objective Map](#)
5. [Assessment Tests](#)
6. [PART I: Exam 1Z0-815, OCP Java SE 11 Programmer I](#)
  1. [Chapter 1: Welcome to Java](#)
    1. [Learning About the Java Environment](#)
    2. [Identifying Benefits of Java](#)
    3. [Understanding the Java Class Structure](#)
    4. [Writing a \*main\(\)\* Method](#)
    5. [Understanding Package Declarations and Imports](#)
    6. [Ordering Elements in a Class](#)
    7. [Code Formatting on the Exam](#)
    8. [Summary](#)
    9. [Exam Essentials](#)
    10. [Review Questions](#)
  2. [Chapter 2: Java Building Blocks](#)
    1. [Creating Objects](#)
    2. [Understanding Data Types](#)
    3. [Declaring Variables](#)
    4. [Initializing Variables](#)
    5. [Managing Variable Scope](#)
    6. [Destroying Objects](#)
    7. [Summary](#)

8. [Exam Essentials](#)
9. [Review Questions](#)
3. [Chapter 3: Operators](#)
  1. [Understanding Java Operators](#)
  2. [Applying Unary Operators](#)
  3. [Working with Binary Arithmetic Operators](#)
  4. [Assigning Values](#)
  5. [Comparing Values](#)
  6. [Making Decisions with the Ternary Operator](#)
  7. [Summary](#)
  8. [Exam Essentials](#)
  9. [Review Questions](#)
4. [Chapter 4: Making Decisions](#)
  1. [Creating Decision-Making Statements](#)
  2. [Writing \*while\* Loops](#)
  3. [Constructing \*for\* Loops](#)
  4. [Controlling Flow with Branching](#)
  5. [Summary](#)
  6. [Exam Essentials](#)
  7. [Review Questions](#)
5. [Chapter 5: Core Java APIs](#)
  1. [Creating and Manipulating Strings](#)
  2. [Using the \*StringBuilder\* Class](#)
  3. [Understanding Equality](#)
  4. [Understanding Java Arrays](#)
  5. [Understanding an \*ArrayList\*](#)
  6. [Creating Sets and Maps](#)
  7. [Calculating with Math APIs](#)
  8. [Summary](#)
  9. [Exam Essentials](#)
  10. [Review Questions](#)
6. [Chapter 6: Lambdas and Functional Interfaces](#)
  1. [Writing Simple Lambdas](#)
  2. [Introducing Functional Interfaces](#)
  3. [Working with Variables in Lambdas](#)

4. [Calling APIs with Lambdas](#)
  5. [Summary](#)
  6. [Exam Essentials](#)
  7. [Review Questions](#)
7. [Chapter 7: Methods and Encapsulation](#)
  1. [Designing Methods](#)
  2. [Working with Varargs](#)
  3. [Applying Access Modifiers](#)
  4. [Applying the \*static\* Keyword](#)
  5. [Passing Data among Methods](#)
  6. [Overloading Methods](#)
  7. [Encapsulating Data](#)
  8. [Summary](#)
  9. [Exam Essentials](#)
  10. [Review Questions](#)
8. [Chapter 8: Class Design](#)
  1. [Understanding Inheritance](#)
  2. [Creating Classes](#)
  3. [Declaring Constructors](#)
  4. [Inheriting Members](#)
  5. [Understanding Polymorphism](#)
  6. [Summary](#)
  7. [Exam Essentials](#)
  8. [Review Questions](#)
9. [Chapter 9: Advanced Class Design](#)
  1. [Creating Abstract Classes](#)
  2. [Implementing Interfaces](#)
  3. [Introducing Inner Classes](#)
  4. [Summary](#)
  5. [Exam Essentials](#)
  6. [Review Questions](#)
10. [Chapter 10: Exceptions](#)
  1. [Understanding Exceptions](#)
  2. [Recognizing Exception Classes](#)
  3. [Handling Exceptions](#)

4. [Calling Methods That Throw Exceptions](#)
  5. [Summary](#)
  6. [Exam Essentials](#)
  7. [Review Questions](#)
11. [Chapter 11: Modules](#)
  1. [Introducing Modules](#)
  2. [Creating and Running a Modular Program](#)
  3. [Updating Our Example for Multiple Modules](#)
  4. [Diving into the \*module-info\* File](#)
  5. [Discovering Modules](#)
  6. [Reviewing Command-Line Options](#)
  7. [Summary](#)
  8. [Exam Essentials](#)
  9. [Review Questions](#)
7. [PART II: Exam 1Z0-816, OCP Java SE 11 Programmer II Exam 1Z0-817, Upgrade OCP Java SE 11](#)
  1. [Chapter 12: Java Fundamentals](#)
    1. [Applying the \*final\* Modifier](#)
    2. [Working with Enums](#)
    3. [Creating Nested Classes](#)
    4. [Understanding Interface Members](#)
    5. [Introducing Functional Programming](#)
    6. [Summary](#)
    7. [Exam Essentials](#)
    8. [Review Questions](#)
  2. [Chapter 13: Annotations](#)
    1. [Introducing Annotations](#)
    2. [Creating Custom Annotations](#)
    3. [Applying Annotations](#)
    4. [Declaring Annotation-Specific Annotations](#)
    5. [Using Common Annotations](#)
    6. [Summary](#)
    7. [Exam Essentials](#)
    8. [Review Questions](#)
  3. [Chapter 14: Generics and Collections](#)

1. [Using Method References](#)
2. [Using Wrapper Classes](#)
3. [Using the Diamond Operator](#)
4. [Using Lists, Sets, Maps, and Queues](#)
5. [Sorting Data](#)
6. [Working with Generics](#)
7. [Summary](#)
8. [Exam Essentials](#)
9. [Review Questions](#)
4. [Chapter 15: Functional Programming](#)
  1. [Working with Built-in Functional Interfaces](#)
  2. [Returning an \*Optional\*](#)
  3. [Using Streams](#)
  4. [Working with Primitive Streams](#)
  5. [Working with Advanced Stream Pipeline Concepts](#)
  6. [Summary](#)
  7. [Exam Essentials](#)
  8. [Review Questions](#)
5. [Chapter 16: Exceptions, Assertions, and Localization](#)
  1. [Reviewing Exceptions](#)
  2. [Creating Custom Exceptions](#)
  3. [Automating Resource Management](#)
  4. [Declaring Assertions](#)
  5. [Working with Dates and Times](#)
  6. [Supporting Internationalization and Localization](#)
  7. [Loading Properties with Resource Bundles](#)
  8. [Summary](#)
  9. [Exam Essentials](#)
  10. [Review Questions](#)
6. [Chapter 17: Modular Applications](#)
  1. [Reviewing Module Directives](#)
  2. [Comparing Types of Modules](#)
  3. [Analyzing JDK Dependencies](#)
  4. [Migrating an Application](#)
  5. [Creating a Service](#)

6. [Summary](#)
7. [Exam Essentials](#)
8. [Review Questions](#)
7. [Chapter 18: Concurrency](#)
  1. [Introducing Threads](#)
  2. [Creating Threads with the Concurrency API](#)
  3. [Writing Thread-Safe Code](#)
  4. [Using Concurrent Collections](#)
  5. [Identifying Threading Problems](#)
  6. [Working with Parallel Streams](#)
  7. [Summary](#)
  8. [Exam Essentials](#)
  9. [Review Questions](#)
8. [Chapter 19: I/O](#)
  1. [Understanding Files and Directories](#)
  2. [Introducing I/O Streams](#)
  3. [Common I/O Stream Operations](#)
  4. [Working with I/O Stream Classes](#)
  5. [Interacting with Users](#)
  6. [Summary](#)
  7. [Exam Essentials](#)
  8. [Review Questions](#)
9. [Chapter 20: NIO.2](#)
  1. [Introducing NIO.2](#)
  2. [Interacting with Paths](#)
  3. [Operating on Files and Directories](#)
  4. [Managing File Attributes](#)
  5. [Applying Functional Programming](#)
  6. [Comparing Legacy java.io.File and NIO.2 Methods](#)
  7. [Summary](#)
  8. [Exam Essentials](#)
  9. [Review Questions](#)
10. [Chapter 21: JDBC](#)
  1. [Introducing Relational Databases and SQL](#)
  2. [Introducing the Interfaces of JDBC](#)

3. [Connecting to a Database](#)
4. [Working with a \*PreparedStatement\*](#)
5. [Getting Data from a ResultSet](#)
6. [Calling a \*CallableStatement\*](#)
7. [Closing Database Resources](#)
8. [Summary](#)
9. [Exam Essentials](#)
10. [Review Questions](#)
11. [Chapter 22: Security](#)
  1. [Designing a Secure Object](#)
  2. [Introducing Injection and Input Validation](#)
  3. [Working with Confidential Information](#)
  4. [Serializing and Deserializing Objects](#)
  5. [Constructing Sensitive Objects](#)
  6. [Preventing Denial of Service Attacks](#)
  7. [Summary](#)
  8. [Exam Essentials](#)
  9. [Review Questions](#)
8. [Appendix: Answers to Review Questions](#)
  1. [Chapter 1: Welcome to Java](#)
  2. [Chapter 2: Java Building Blocks](#)
  3. [Chapter 3: Operators](#)
  4. [Chapter 4: Making Decisions](#)
  5. [Chapter 5: Core Java APIs](#)
  6. [Chapter 6: Lambdas and Functional Interfaces](#)
  7. [Chapter 7: Methods and Encapsulation](#)
  8. [Chapter 8: Class Design](#)
  9. [Chapter 9: Advanced Class Design](#)
  10. [Chapter 10: Exceptions](#)
  11. [Chapter 11: Modules](#)
  12. [Chapter 12: Java Fundamentals](#)
  13. [Chapter 13: Annotations](#)
  14. [Chapter 14: Generics and Collections](#)
  15. [Chapter 15: Functional Programming](#)
  16. [Chapter 16: Exceptions, Assertions, and Localization](#)

- 17. [Chapter 17: Modular Applications](#)
- 18. [Chapter 18: Concurrency](#)
- 19. [Chapter 19: I/O](#)
- 20. [Chapter 20: NIO.2](#)
- 21. [Chapter 21: JDBC](#)
- 22. [Chapter 22: Security](#)
- 9. [Index](#)
- 10. [Online Test Bank](#)
  - 1. [Register and Access the Online Test Bank](#)
- 11. [End User License Agreement](#)

## List of Tables

- 1. Introduction
  - 1. [TABLE I.1 Exam information](#)
- 2. Chapter 1
  - 1. [TABLE 1.1 Running programs](#)
  - 2. [TABLE 1.2 Setup procedure by operating system](#)
  - 3. [TABLE 1.3 Options you need to know for the exam: javac](#)
  - 4. [TABLE 1.4 Options you need to know for the exam: java](#)
  - 5. [TABLE 1.5 Options you need to know for the exam: jar](#)
  - 6. [TABLE 1.6 Order for declaring a class](#)
- 3. Chapter 2
  - 1. [TABLE 2.1 Primitive types](#)
  - 2. [TABLE 2.2 Reserved words](#)
  - 3. [TABLE 2.3 Default initialization values by type](#)
  - 4. [TABLE 2.4 Tracking scope by block](#)
- 4. Chapter 3
  - 1. [TABLE 3.1 Order of operator precedence](#)
  - 2. [TABLE 3.2 Unary operators](#)
  - 3. [TABLE 3.3 Binary arithmetic operators](#)
  - 4. [TABLE 3.4 Simple assignment operator](#)
  - 5. [TABLE 3.5 Compound assignment operators](#)
  - 6. [TABLE 3.6 Equality operators](#)
  - 7. [TABLE 3.7 Relational operators](#)



8. [TABLE 3.8 Logical operators](#)
9. [TABLE 3.9 Short-circuit operators](#)
5. Chapter 4
  1. [TABLE 4.1 Advanced flow control usage](#)
6. Chapter 5
  1. [TABLE 5.1 Binary search rules](#)
  2. [TABLE 5.2 Arrays.compare\(\). examples](#)
  3. [TABLE 5.3 Equality vs. comparison vs. mismatch](#)
  4. [TABLE 5.4 Wrapper classes](#)
  5. [TABLE 5.5 Converting from a String](#)
  6. [TABLE 5.6 Array and list conversions](#)
  7. [TABLE 5.7 Common Map methods](#)
7. Chapter 6
  1. [TABLE 6.1 Valid lambdas](#)
  2. [TABLE 6.2 Invalid lambdas that return boolean](#)
  3. [TABLE 6.3 Basic functional interfaces](#)
  4. [TABLE 6.4 Rules for accessing a variable from a lambda body inside a method](#)
8. Chapter 7
  1. [TABLE 7.1 Parts of a method declaration](#)
  2. [TABLE 7.2 Access modifiers](#)
  3. [TABLE 7.3 Static vs. instance calls](#)
  4. [TABLE 7.4 The order that Java uses to choose the right overloaded method](#)
  5. [TABLE 7.5 Naming conventions for getters and setters](#)
9. Chapter 10
  1. [TABLE 10.1 Types of exceptions and errors](#)
  2. [TABLE 10.2 Legal vs. illegal configurations with a traditional try statement](#)
  3. [TABLE 10.3 Legal vs. illegal configurations with a try-with-resources statement](#)
10. Chapter 11
  1. [TABLE 11.1 Options you need to know for using modules with javac](#)

2. [TABLE 11.2 Options you need to know for using modules with java](#)
3. [TABLE 11.3 Access control with modules](#)
4. [TABLE 11.4 Modes using jmod](#)
5. [TABLE 11.5 Comparing command-line operations](#)
6. [TABLE 11.6 Options you need to know for the exam: javac](#)
7. [TABLE 11.7 Options you need to know for the exam: java](#)
8. [TABLE 11.8 Options you need to know for the exam: jar](#)
9. [TABLE 11.9 Options you need to know for the exam: jdeps](#)

## 11. Chapter 12

1. [TABLE 12.1 Modifiers in nested classes](#)
2. [TABLE 12.2 Members in nested classes](#)
3. [TABLE 12.3 Nested class access rules](#)
4. [TABLE 12.4 Interface member types](#)
5. [TABLE 12.5 Interface member access](#)

## 12. Chapter 13

1. [TABLE 13.1 Values for the @Target annotation](#)
2. [TABLE 13.2 Values for the @Retention annotation](#)
3. [TABLE 13.3 Annotation-specific annotations](#)
4. [TABLE 13.4 Common @SuppressWarnings values](#)
5. [TABLE 13.5 Understanding common annotations](#)
6. [TABLE 13.6 Applying common annotations](#)

## 13. Chapter 14

1. [TABLE 14.1 Functional interfaces used in this chapter](#)
2. [TABLE 14.2 Method references](#)
3. [TABLE 14.3 Wrapper classes](#)
4. [TABLE 14.4 Factory methods to create a List](#)
5. [TABLE 14.5 List methods](#)
6. [TABLE 14.6 Queue methods](#)
7. [TABLE 14.7 Map methods](#)
8. [TABLE 14.8 Behavior of the merge\(\) method](#)
9. [TABLE 14.9 Java Collections Framework types](#)
10. [TABLE 14.10 Collection attributes](#)
11. [TABLE 14.11 Comparison of Comparable and Comparator](#)
12. [TABLE 14.12 Helper static methods for building a Comparator](#)

13. [TABLE 14.13 Helper default methods for building a Comparator](#)
14. [TABLE 14.14 Types of bounds](#)
15. [TABLE 14.15 Why we need a lower bound](#)
14. Chapter 15
  1. [TABLE 15.1 Common functional interfaces](#)
  2. [TABLE 15.2 Convenience methods](#)
  3. [TABLE 15.3 Optional instance methods](#)
  4. [TABLE 15.4 Intermediate vs. terminal operations](#)
  5. [TABLE 15.5 Creating a source](#)
  6. [TABLE 15.6 Terminal stream operations](#)
  7. [TABLE 15.7 Common primitive stream methods](#)
  8. [TABLE 15.8 Mapping methods between types of streams](#)
  9. [TABLE 15.9 Function parameters when mapping between types of streams](#)
  10. [TABLE 15.10 Optional types for primitives](#)
  11. [TABLE 15.11 Common functional interfaces for primitives](#)
  12. [TABLE 15.12 Primitive-specific functional interfaces](#)
  13. [TABLE 15.13 Examples of grouping/partitioning collectors](#)
15. Chapter 16
  1. [TABLE 16.1 Unchecked exceptions](#)
  2. [TABLE 16.2 Checked exceptions](#)
  3. [TABLE 16.3 Assertion applications](#)
  4. [TABLE 16.4 Date and time types](#)
  5. [TABLE 16.5 Common date/time symbols](#)
  6. [TABLE 16.6 Supported date/time symbols](#)
  7. [TABLE 16.7 Factory methods to get a `NumberFormat`](#)
  8. [TABLE 16.8 `DecimalFormat` symbols](#)
  9. [TABLE 16.9 Factory methods to get a `DateTimeFormatter`](#)
  10. [TABLE 16.10 `Locale.Category` values](#)
  11. [TABLE 16.11 Picking a resource bundle for French/France with default locale E...](#)
  12. [TABLE 16.12 Selecting resource bundle properties](#)
16. Chapter 17
  1. [TABLE 17.1 Common module directives](#)
  2. [TABLE 17.2 Practicing with automatic module names](#)

3. [TABLE 17.3 Properties of modules types](#)
4. [TABLE 17.4 Common modules](#)
5. [TABLE 17.5 Java modules prefixed with java](#)
6. [TABLE 17.6 Java modules prefixed with jdk](#)
7. [TABLE 17.7 Comparing migration strategies](#)
8. [TABLE 17.8 Reviewing services](#)

## 17. Chapter 18

1. [TABLE 18.1 ExecutorService methods](#)
2. [TABLE 18.2 Future methods](#)
3. [TABLE 18.3 TimeUnit values](#)
4. [TABLE 18.4 ScheduledExecutorService methods](#)
5. [TABLE 18.5 Executors factory methods](#)
6. [TABLE 18.6 Atomic classes](#)
7. [TABLE 18.7 Common atomic methods](#)
8. [TABLE 18.8 Lock methods](#)
9. [TABLE 18.9 Concurrent collection classes](#)
10. [TABLE 18.10 BlockingQueue waiting methods](#)
11. [TABLE 18.11 Synchronized collections methods](#)

## 18. Chapter 19

1. [TABLE 19.1 Commonly used java.io.File methods](#)
2. [TABLE 19.2 The java.io abstract stream base classes](#)
3. [TABLE 19.3 The java.io concrete stream classes](#)
4. [TABLE 19.4 Common I/O stream methods](#)
5. [TABLE 19.5 Common print stream `format\(\)` symbols](#)

## 19. Chapter 20

1. [TABLE 20.1 File system symbols](#)
2. [TABLE 20.2 Common NIO.2 method arguments](#)
3. [TABLE 20.3 Path methods](#)
4. [TABLE 20.4 Files methods](#)
5. [TABLE 20.5 The attributes and view types](#)
6. [TABLE 20.6 Walking a directory with a cycle using breadth-first search](#)
7. [TABLE 20.7 Comparison of `java.io.File` and NIO.2 methods](#)

## 20. Chapter 21

1. [TABLE 21.1 CRUD operations](#)

2. [TABLE 21.2 SQL runnable by the execute method](#)
3. [TABLE 21.3 Return types of execute methods](#)
4. [TABLE 21.4 PreparedStatement methods](#)
5. [TABLE 21.5 ResultSet get methods](#)
6. [TABLE 21.6 Sample stored procedures](#)
7. [TABLE 21.7 Stored procedure parameter types](#)

## 21. Chapter 22

1. [TABLE 22.1 Types of confidential data](#)
2. [TABLE 22.2 Methods for serialization and deserialization](#)

## List of Illustrations

### 1. Introduction

1. [FIGURE I.1 Past and current Java certifications](#)
2. [FIGURE I.2 Latest Java certification exams](#)
3. [FIGURE I.3 Exam prerequisites](#)

### 2. Chapter 1

1. [FIGURE 1.1 Compiling with packages](#)
2. [FIGURE 1.2 Compiling with packages and directories](#)

### 3. Chapter 2

1. [FIGURE 2.1 An object in memory can be accessed only via a reference.](#)
2. [FIGURE 2.2 Your drawing after line 5](#)
3. [FIGURE 2.3 Your drawing after line 7](#)

### 4. Chapter 3

1. [FIGURE 3.1 The logical truth tables for &, |, and ^](#)

### 5. Chapter 4

1. [FIGURE 4.1 The structure of an if statement](#)
2. [FIGURE 4.2 The structure of an else statement](#)
3. [FIGURE 4.3 The structure of a switch statement](#)
4. [FIGURE 4.4 The structure of a while statement](#)
5. [FIGURE 4.5 The structure of a do/while statement](#)
6. [FIGURE 4.6 The structure of a basic for loop](#)
7. [FIGURE 4.7 The structure of an enhanced for-each loop](#)
8. [FIGURE 4.8 The structure of a break statement](#)

9. [FIGURE 4.9 The structure of a continue statement](#)

6. Chapter 5

1. [FIGURE 5.1 Indexing for a string](#)
2. [FIGURE 5.2 Indexes for a substring](#)
3. [FIGURE 5.3 The basic structure of an array](#)
4. [FIGURE 5.4 An empty array](#)
5. [FIGURE 5.5 An initialized array](#)
6. [FIGURE 5.6 An array pointing to strings](#)
7. [FIGURE 5.7 A sparsely populated multidimensional array](#)
8. [FIGURE 5.8 An asymmetric multidimensional array](#)
9. [FIGURE 5.9 Example of a Set](#)
10. [FIGURE 5.10 Example of a Map](#)

7. Chapter 6

1. [FIGURE 6.1 Lambda syntax omitting optional parts](#)
2. [FIGURE 6.2 Lambda syntax, including optional parts](#)

8. Chapter 7

1. [FIGURE 7.1 Method declaration](#)
2. [FIGURE 7.2 Classes used to show private and default access](#)
3. [FIGURE 7.3 Classes used to show protected access](#)
4. [FIGURE 7.4 Copying a reference with pass-by-value](#)

9. Chapter 8

1. [FIGURE 8.1 Types of inheritance](#)
2. [FIGURE 8.2 Java object inheritance](#)
3. [FIGURE 8.3 Defining and extending a class](#)
4. [FIGURE 8.4 Object vs. reference](#)

10. Chapter 9

1. [FIGURE 9.1 Defining an interface](#)
2. [FIGURE 9.2 Implementing an interface](#)
3. [FIGURE 9.3 Interface Inheritance](#)

11. Chapter 10

1. [FIGURE 10.1 Categories of exception](#)
2. [FIGURE 10.2 The syntax of a try statement](#)
3. [FIGURE 10.3 The syntax of a multi-catch block](#)
4. [FIGURE 10.4 The syntax of a try statement with finally](#)
5. [FIGURE 10.5 The syntax of a basic try-with-resources](#)

6. [FIGURE 10.6 The syntax of try-with-resources including catch/finally](#)

7. [FIGURE 10.7 A method stack](#)

## 12. Chapter 11

1. [FIGURE 11.1 Design of a modular system](#)

2. [FIGURE 11.2 Looking inside a module](#)

3. [FIGURE 11.3 Contents of zoo.animal.feeding](#)

4. [FIGURE 11.4 Module zoo.animal.feeding directory structure](#)

5. [FIGURE 11.5 Running a module using java](#)

6. [FIGURE 11.6 Module zoo.animal.feeding directory structure with class and jar fil...](#)

7. [FIGURE 11.7 Modules depending on zoo.animal.feeding](#)

8. [FIGURE 11.8 Contents of zoo.animal.care](#)

9. [FIGURE 11.9 Module zoo.animal.care directory structure](#)

10. [FIGURE 11.10 Dependencies for zoo.animal.talks](#)

11. [FIGURE 11.11 Contents of zoo.animal.talks](#)

12. [FIGURE 11.12 Contents of zoo.staff](#)

13. [FIGURE 11.13 Dependencies for zoo.staff](#)

14. [FIGURE 11.14 Transitive dependency version of our modules](#)

## 13. Chapter 12

1. [FIGURE 12.1 Lambda syntax omitting optional parts](#)

2. [FIGURE 12.2 Lambda syntax, including optional parts](#)

## 14. Chapter 13

1. [FIGURE 13.1 Annotation declaration](#)

2. [FIGURE 13.2 Using an annotation](#)

## 15. Chapter 14

1. [FIGURE 14.1 The `Collection` interface is the root of all collections except m...](#)

2. [FIGURE 14.2 Example of a `List`](#)

3. [FIGURE 14.3 Example of a `Set`](#)

4. [FIGURE 14.4 Examples of a `HashSet` and `TreeSet`](#)

5. [FIGURE 14.5 Example of a `Queue`](#)

6. [FIGURE 14.6 Working with a queue](#)

7. [FIGURE 14.7 Example of a `Map`](#)

## 16. Chapter 15

1. [FIGURE 15.1 Optional](#)
2. [FIGURE 15.2 Stream pipeline](#)
3. [FIGURE 15.3 Steps in running a stream pipeline](#)
4. [FIGURE 15.4 A stream pipeline with a limit](#)
5. [FIGURE 15.5 Stream pipeline with multiple intermediate operations](#)

## 17. Chapter 16

1. [FIGURE 16.1 The syntax of a `try` statement](#)
2. [FIGURE 16.2 The syntax of a `try-with-resources` statement](#)
3. [FIGURE 16.3 Categories of exceptions](#)
4. [FIGURE 16.4 The syntax of `assert` statements](#)
5. [FIGURE 16.5 `Locale` formats](#)

## 18. Chapter 17

1. [FIGURE 17.1 A named module](#)
2. [FIGURE 17.2 An automatic module](#)
3. [FIGURE 17.3 An unnamed module](#)
4. [FIGURE 17.4 Determining the order](#)
5. [FIGURE 17.5 Determining the order when not unique](#)
6. [FIGURE 17.6 Bottom-up migration](#)
7. [FIGURE 17.7 Top-down migration](#)
8. [FIGURE 17.8 First attempt at decomposition](#)
9. [FIGURE 17.9 Removing the cyclic dependencies](#)
10. [FIGURE 17.10 Modules in the tour application](#)

## 19. Chapter 18

1. [FIGURE 18.1 Process model](#)
2. [FIGURE 18.2 `ExecutorService` life cycle](#)
3. [FIGURE 18.3 Lack of thread synchronization](#)
4. [FIGURE 18.4 Thread synchronization using atomic operations](#)
5. [FIGURE 18.5 Race condition on user creation](#)

## 20. Chapter 19

1. [FIGURE 19.1 Directory and file hierarchy](#)
2. [FIGURE 19.2 Visual representation of a stream](#)
3. [FIGURE 19.3 Serialization process](#)
4. [FIGURE 19.4 Diagram of I/O stream classes](#)

## 21. Chapter 20

1. [FIGURE 20.1 File system with a symbolic link](#)



2. [FIGURE 20.2 NIO.2 class and interface relationships](#)
3. [FIGURE 20.3 Relative paths using path symbols](#)
4. [FIGURE 20.4 Comparing file uniqueness](#)
5. [FIGURE 20.5 File and directory as a tree structure](#)
6. [FIGURE 20.6 File system with cycle](#)

## 22. Chapter 21

1. [FIGURE 21.1 Tables in our relational database](#)
2. [FIGURE 21.2 Key JDBC interfaces](#)
3. [FIGURE 21.3 The JDBC URL format](#)
4. [FIGURE 21.4 Types of statements](#)
5. [FIGURE 21.5 The `ResultSet` cursor](#)

## 23. Chapter 22

1. [FIGURE 22.1 `Cloneable` logic](#)
2. [FIGURE 22.2 Hours table](#)
3. [FIGURE 22.3 Directory structure](#)
4. [FIGURE 22.4 Writing and reading an employee](#)

[Support](#)   [Sign Out](#)