

Java Full Throttle Script

Items in **red** are likely to be skipped in the presentation due to time constraints, but are provided for your reference. Additional items might be skipped during today's class based on the number of questions from the audience.

Everything is discussed in detail in both my [Java 8 and 9 Fundamentals LiveLessons videos](#) and my [Java How to Program, 11/e textbook](#) here on Safari.

Introduction

- Setup: https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_BYB01

Chapter 1: Getting Started

- **SLIDE 13: Popular IDEs**
- Test-drive: https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson01_intro
- <https://stackoverflow.com/questions/21535023/how-to-get-your-netbeans-project-into-eclipse>

Chapter 2: Intro to Java

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson02_intro
- **SLIDE 16: Overview**
- Fig. 2.1 Welcome
- Fig. 2.7 Addition
- **SLIDE 17: Arithmetic operators**
- **Project 05: Java 15 text blocks**

Chapter 3: Intro to Classes

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson03_intro
- **SLIDE 18: Overview**
- Fig. 3.8-3.9 Account class and test program
- **SLIDES 19-21: Primitive and reference types**

Chapter 4: Control Statements 1

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson04_intro

- **SLIDE 22: Overview**
- **SLIDE 23: Relational/Equality operators**
- Launch JShell in verbose mode for extra info
jshell -v (verbose)
or
/set feedback verbose (internal)
- **JSHELL: Snippets for**
 - If
 - If/else
 - Nested if/else
 - **SLIDE 24** Conditional operator
 - while
- **SLIDE 25:** Explicitly converting between types
 - **JSHELL:** Show cast operator in JShell
- **SLIDE 26:** Compound assignment operators
 - **JSHELL:** Use one in JShell
- **SLIDE 27:** Increment/decrement operators
 - **JSHELL:** Use in JShell

Chapter 5: Control Statements 2

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson05_intro
- **SLIDE 28: Overview**
- **JSHELL:** for loop snippets
- **JSHELL:** do/while snippet
- **JSHELL:** switch—cases can be strings or enum constants too.
- Fig. 5.13 break
- Fig. 5.14 continue
- Fig. 5.19 logical operators

Chapter 6: Methods

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson06_intro
- **SLIDE 29: Overview**
- **SLIDES 30-32:** Discuss static class members in slides
 - **JShell Math class**
- **SLIDE 33:** Why is main static?

- **SLIDE 34:** Argument promotion
- Java API
 - <https://docs.oracle.com/javase/10/docs/api/overview-summary.html>
- **Craps Program**
 - **SLIDE 35:** enum types
 - Fig. 6.8 Craps (just the enum)
 - **SLIDE 36:** Why Some Constants Are Not Defined as enum Constants
- Fig. 6.9 Scope
 - **SLIDE 37:** Scope of Declarations
- Fig. 6.10 Overloading

Chapter 7: Arrays and ArrayLists

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson07_intro
- **SLIDE 38: Overview**
- Fig. 7.2 array of 0s
- Fig. 7.3 array with initializer list
- Fig. 7.9-11: Card Shuffling
- **SLIDE 39:** Pass-By-Value vs. Pass-By-Reference
- Fig. 7.17 multidimensional arrays
- **Class Arrays** for common array manipulations:
 - <https://docs.oracle.com/javase/10/docs/api/java/util/Arrays.html>
- **SLIDES 40:** ArrayList
- Fig. 7.24 ArrayList

Chapter 8: Classes—A Deeper Look

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson08_intro
- **SLIDE 41: Overview**
- Fig. 8.5-8.6: Time with overloaded constructors.
- Project 22: Java 14 records
- **SLIDES 42-43:** Garbage collection
- **(IF WE HAVE TIME) SLIDES 44-45:** enum Types revisited
 - Fig. 8.10-11 enum

Chapter 9: Inheritance

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson09_intro
- **SLIDE 46: Overview**
- Show Object class docs
- **SLIDES 47-48:** protected members

Chapter 10: Polymorphism

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson10_intro
- **SLIDE 49: Overview**
- **SLIDE 50:** Employee Hierarchy
- **SLIDE 51:** earnings and toString
- Fig. 10.4-9: Employee hierarchy
- Project 23_b: Java 14's pattern matching instanceof
- **SLIDES 52-53** final Methods and Classes.
- **SLIDE 54:** Payable hierarchy
- Fig. 10.11-14 Payable Hierarchy
- **SLIDES 55-59** Common Interfaces
- **SLIDES 60-65:** Interface enhancements
- **SLIDES 66-67:** private Constructors

Chapter 11: Exception Handling

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson10_intro
- **SLIDE 68: Overview**
- Fig. 11.3 exception handling
- **SLIDE 69:** Exception hierarchy
- Fig. 11.5 finally
- **SLIDES 70-71:** Try with resources

Chapter 12: JavaFX 1

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson22_intro
- **SLIDE 72: Overview**
- Java FX not in JDK 11+.

- Instructions for getting started
 - <https://openjfx.io/openjfx-docs/>
- Downloading the JavaFX Runtime
 - <https://gluonhq.com/products/javafx/>
- PATH_TO_FX Environment Variable (these depend on where you place the SDK on your system and which SDK version you have)
 - Mac/Linux
 - export PATH_TO_FX=/Users/pauldeitel/Downloads/javafx-sdk-14/lib
 - Windows
 - set PATH_TO_FX="c:\Users\pauldeitel\Downloads\javafx-sdk-14\lib"
- Compiling Painter project at the command line (Windows users replace \$PATH_TO_FX with %PATH_TO_FX%)
 - javac --module-path \$PATH_TO_FX --add-modules=javafx.controls,javafx.graphics,javafx.fxml *.java
- Running Painter project at the command line (Windows users replace \$PATH_TO_FX with %PATH_TO_FX%)
 - java --module-path \$PATH_TO_FX --add-modules=javafx.controls,javafx.graphics,javafx.fxml Painter
- **SLIDE 73**
- Tip Calculator app

Chapter 13: JavaFX 2

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson23_intro
- Color chooser app demo
- CoverViewer app demo
- CoverViewer app demo

Chapter 16: Generic Collections

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson16_intro
- **SLIDE 75: Overview**
- **SLIDE 76:** Intro and table of basic collection interfaces
- **SLIDES 77-78/JSHELL:** Show type wrapper classes in JShell
- **SLIDE 79:** Show boxing/unboxing in JShell
- **SLIDE 80:** When discussing interface Collection, show online docs:

- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Collection.html>
- <https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/util/Collections.html>
- Fig. 16.6-8 sorting lists
- (IF WE HAVE TIME) Fig. 16.15 HashSet
- (IF WE HAVE TIME) Fig. 16.17 HashMap
- **SLIDE 81:** Fig. 16.20 Convenience factory methods.

Chapter 17: Lambdas and Streams

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson17_intro
- **SLIDE 82-83: Overview**
- **SLIDES 84-86:** External iteration
- **SLIDES 87-88:** What not how
- Fig. 17.3 sum/reduction/internal iteration
- **SLIDES 89-91:** then Fig. 17.4 map/reduce
- **SLIDES 92-94:** Lambdas
- **SLIDES 95-97:** Intermediate/terminal operations
- **SLIDES 98-99** then Fig. 17.7: Filter map reduce
- **SLIDES 100-101:** How Elements move through stream
 - Use code snippet in copy/paste to show how by modifying 17.7
- **(IF TIME) SLIDE 102** then Fig. 17.8: Method references
 - Follow with additional **SLIDES 103-104 and 105**
- **SLIDES 106-111** Functional Interfaces
- Fig. 17.13-21 Employee example
 - **SLIDES 112-113** Composing Lambdas
- **(IF TIME) SLIDES 114-115 then TipCalculator with lambda event handler**

Chapter 23: Concurrent Processing and Parallel Streams

- https://learning.oreilly.com/videos/java-8-fundamentals/9780133489354/9780133489354-JFUN_lesson20_intro
- **SLIDE 116-118: Overview**
- **SLIDE 128-130: Overview of Executors**
- Fig. 23.3-23.4 Executor/Runnables
- Fig. 23.29 sort/parallelSort

- Fig. 23.30 sequential vs. parallel streams

What's New in Java 10 – 16