
ORACLE®
ACADEMY

Java Foundations

1-1
About This Course

ORACLE ACADEMY

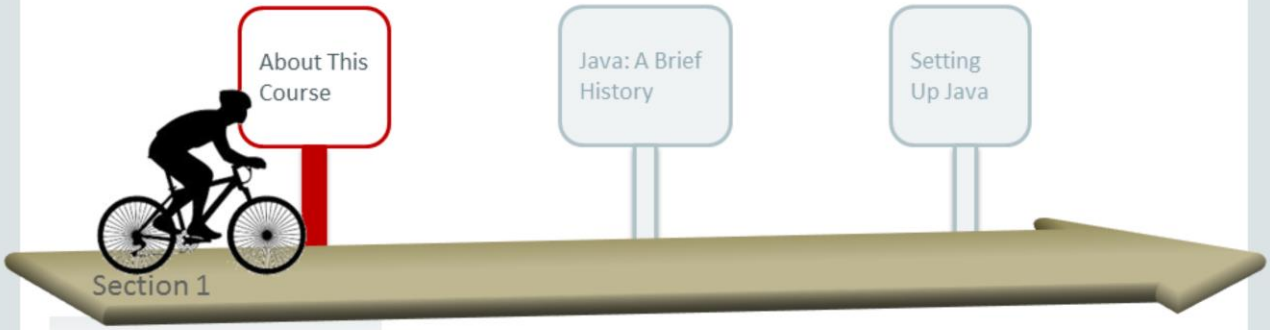
Objectives

This lesson covers the following objectives:

- Identify course goals and objectives
- Understand the course environment
- Describe the course learning strategy

Topics

- Course Overview
- Course Learning Strategy
- After This Course



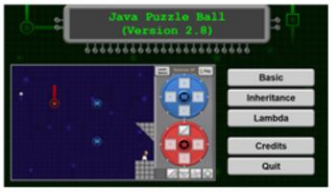
Course Goals

- Demonstrate knowledge of basic programming language concepts
- Demonstrate knowledge of the Java programming language
- Implement basic Java programming and object-oriented concepts

Course Tools

- NetBeans
  - A tool used for writing code.
  - Installation instructions are provided later.

- Java Puzzle Ball
  - A game used throughout the course.
  - It's available through Oracle iLearning as soon as the relevant lessons occur.



- Other materials are available from Oracle iLearning as the relevant lessons occur.

The lead developer of this course was also the lead developer of Java Puzzle Ball.

Your Code

- You'll type your code in NetBeans.
- NetBeans code looks like this:

```
2    public class HelloWorld {
3
4        public static void main(String[] args) {
5            System.out.println("Hello World!");
6        }
7    }
```

Course Outline

- **Section 1**: What Is Java?
  - Lesson 1: About This Course
  - Lesson 2: Java: A Brief History
  - Lesson 3: Setting Up Java
- **Section 2**: Java Basics
  - Lesson 1: The Software Development Process
  - Lesson 2: What Is My Program Doing?
  - Lesson 3: Introduction to Object-Oriented Programming Concepts

Course Outline

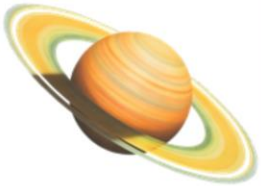
- **Section 3**: Data Types
  - Lesson 1: What Is a Variable?
  - Lesson 2: Numeric Data Types
  - Lesson 3: Textual Data Types
  - Lesson 4: Converting Between Data Types
  - Lesson 5: Keyboard Input

Course Outline

- **Section 4**: Important Classes from the Java API
  - Lesson 1: What Is a Method?
  - Lesson 2: The Import Declaration and Packages
  - Lesson 3: The `String` Class
  - Lesson 4: The `Random` Class
  - Lesson 5: The `Math` Class
- **Section 5**: Decision Statements
  - Lesson 1: Boolean Expressions and `if/else` Constructs
  - Lesson 2: Conditional Execution
  - Lesson 3: `switch` Statements

Course Outline

- **Section 6**: Loop Statements
  - Lesson 1: `for` Loops
  - Lesson 2: `while` and `do/while` Loops
  - Lesson 3: `break` and `continue` Statements
- **Section 7**: Classes
  - Lesson 1: Creating a Class
  - Lesson 2: Instantiating Objects
  - Lesson 3: Overloading Methods
  - Lesson 4: Object Interaction and Encapsulation
  - Lesson 5: `static` Variables and Methods

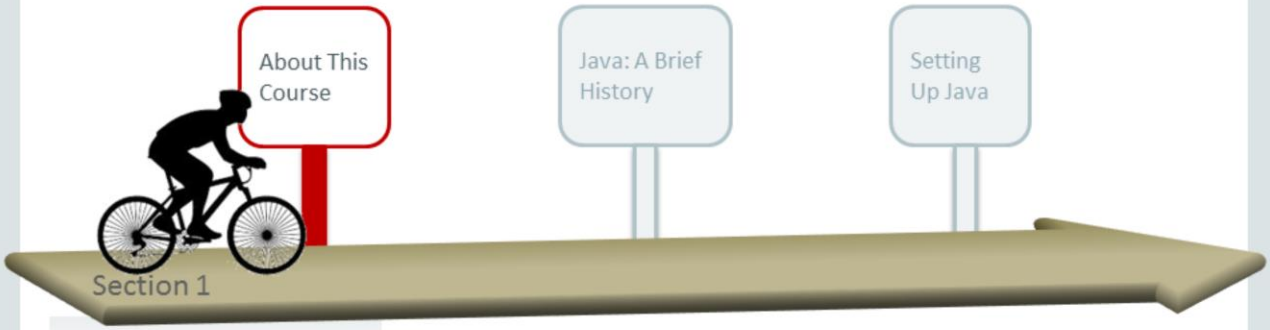


Course Outline

- **Section 8**: Arrays, ArrayLists, and Exceptions
  - Lesson 1: Arrays
  - Lesson 2: ArrayLists
  - Lesson 3: Exception Handling
  - Lesson 4: Debugging Concepts
- **Section 9**: GUI Applications with JavaFX
  - Lesson 1: Introduction to JavaFX
  - Lesson 2: Colors and Shapes
  - Lesson 3: Graphics, Audio, and Mouse Events
- **Section 10**: Final Project

Topics

- Course Overview
- Course Learning Strategy
- After This Course



The Challenges of Designing This Course

- How do we design a Java course that will engage and educate a young audience?
  - Secondary schools
  - Two-year colleges
  - Four-year colleges
- How do we explain technical concepts to an audience that may have no computer science background?
  - This is an introductory programming course.
  - It's designed to be taken with **no prerequisite** courses.
  - But an understanding of basic algebra may be beneficial.

How We Won't Engage You

- Relying on slides that contain walls of text.
- Bombarding you with information without giving you a chance to program.
- Providing technobabble notes that don't explain or match the slides.
- Forcing you to recall obscure technical facts.

How We'll Engage You

- In order for this course to connect with today's youth, its content must be more like you:
  - Hip
  - Trendy
  - Social
- We'll accomplish this with:
  - Backward hats
  - Sunglasses
  - Bling
  - Hashtags

How We'll Really Engage You

- The previous slide is satirical, but it also illustrates the conversational writing style of this course.
- Lessons are written to prepare you to develop software. This is done by:
  - Illustrating core concepts you'll need to create software.
  - Providing code examples that you can reference and build from.
  - Giving you a chance to program and discover solutions.
  - Explaining why certain things are so.
  - Maintaining continuity with scenarios throughout each section. Icons in the "Course Outline" slides hint at these scenarios.
- We want you to feel good about your coding abilities.

Course Components

- Course components include:
  - Lessons
  - Small exercises
  - Quizzes
  - Practices
- All lesson materials should be available through Oracle iLearning.
- This course also includes mid-term and final multiple-choice exams.

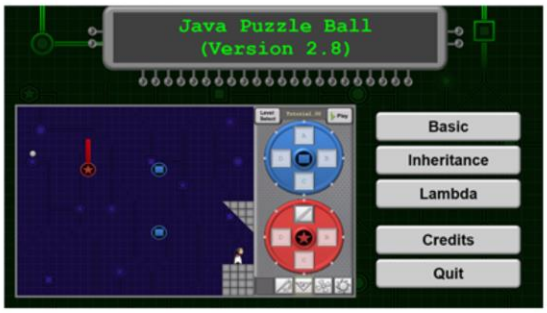
The Problem with Long Lectures

- Students told us that long lectures made them feel exhausted, confused, and helpless when it was finally time to code.

- Most learning occurred when students were allowed to code.
  - The goal of this course is to be hands-on and project-based.
  - That's why we include several smaller exercises throughout each lesson.
  - We also found it counterproductive to bunch exercises at the end of each lesson.

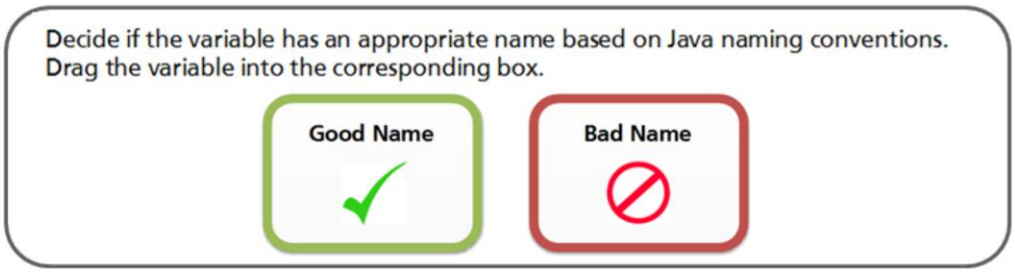
Small Exercises

- Every lesson (starting with Lesson 1-3) includes small exercises that are designed to:
  - Establish familiarity with a concept.
  - Build confidence as concepts are introduced.
  - Build on existing knowledge.
  - Allow you to experiment.
- Exercises include:
  - Listing ideas.
  - Playing games.
  - Editing code.



Quizzes

- Simple quizzes are included at the end of each lesson.
- Quizzes should very closely reflect their lecture.
- They're designed to reinforce key concepts.
- Students told us that they found this approach to be very helpful in other courses.



Quizzes in Oracle iLearning

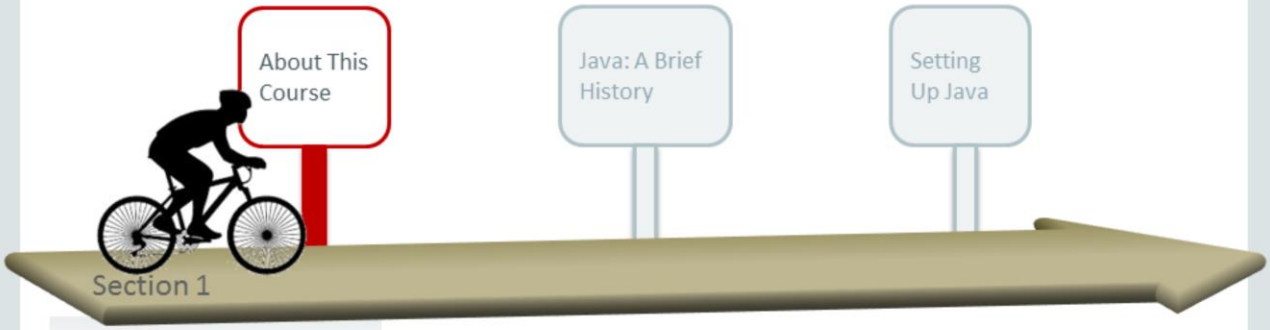
- iLearning quizzes are included at the end of each section.
- iLearning quizzes:
  - will check your knowledge of each section.
  - will prepare you for your midterm and final exams.
  - are designed to reinforce key concepts.

Practices

- Every section includes a practices.
- These are large exercises that should require several hours to complete.
- They describe the features that a particular piece of software requires.
- It's up to you to figure out how to implement these requirements.
- Lessons are designed to explain everything you'll need to find a solution.

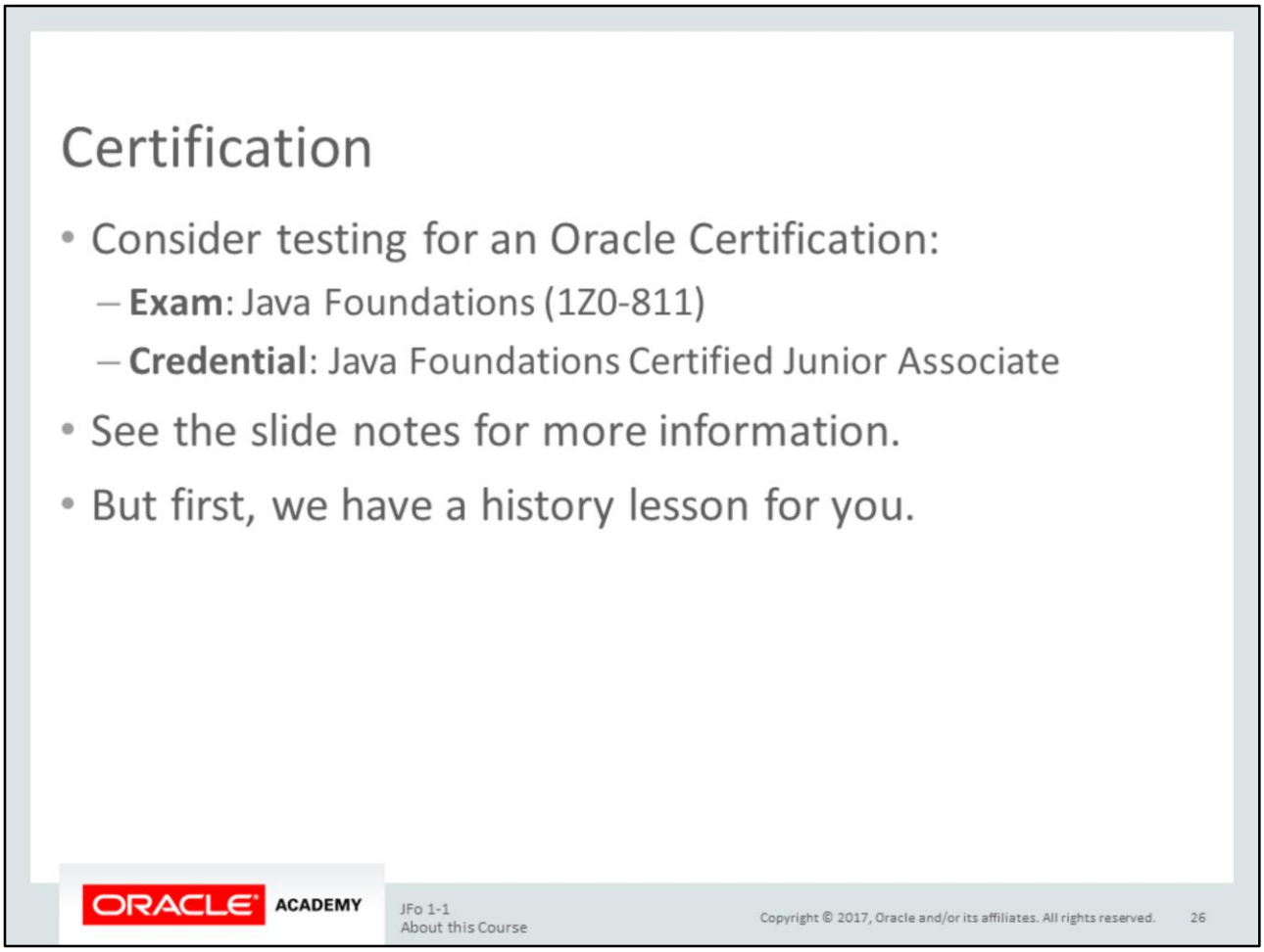
Topics

- Course Overview
- Course Learning Strategy
- After This Course



What You Could Do After Taking This Course

- Enjoy creating simple programs.
- Appreciate an engineer's perspective.
- Take more advanced courses:
  - Oracle Academy Java Fundamentals
  - Oracle University Java SE 8 Fundamentals
  - Oracle University Java SE 8 Programming
  - Advanced Placement Computer Science



Oracle Certification exams are multiple choice. Exams are managed by Oracle University, not Oracle Academy. Of all the certifications that Oracle offers, the Java Foundations exam most closely reflects the concepts you'll learn in this course, but the two don't align perfectly. You'll most likely need further studying to prepare for the certification exam.

Summary

In this lesson, you should have learned how to:

- Identify course goals and objectives
- Understand the course environment
- Describe the course learning strategy


ORACLE
ACADEMY