SLIDE SCRIPT

Slide Script

Every slide in this lesson has speaker notes. They say what to point to, what to say out loud and what answers to listen for. This script prints the same notes, day by day and slide by slide, so you can read ahead the night before or teach from paper.

Where to find the speaker notes

1. In PowerPoint, the notes sit in the box under each slide. If you can't see it, click Notes in the bar at the bottom of the window.
2. When you present, use Presenter View. You see the notes on your laptop, and students see only the slide.
3. In Google Slides, the notes sit in the box under each slide. If you can't see it, open the View menu and choose Show speaker notes.
4. Say: marks words to read out loud, and Ask: marks a question for the class. Every other line tells you what to do.

The lesson cover

Slide 1. Lesson cover

- **Say:** This lesson is Your First Java Program. It runs 3 days.
- **Ask:** How is a Java program built, and how does Java check it before it runs?

Day 1 — Every Java Program Lives in a Class

Slide 2. Day 1 divider: Every Java Program Lives in a Class

- **Say:** Today you will learn the parts of a Java program: a class, a `main` method where Java starts, and statements that print text with `System.out.println()` and `System.out.print()`.

Slide 3. Print a line of text.

- **Say:** You have written code before, so you know this idea. A program shows a line of text on the screen.
- Point to the Java side. **Say:** Java's print line is longer. It starts with System dot out, and it ends with a semicolon.
- **Say:** Both programs print the same line. Today you'll see where this Java line lives in a whole program.

Slide 4. Every Java program lives in a class.

- **Say:** This is a whole Java program. It's the same print line, with two parts wrapped around it.
- Point to line 1. **Say:** Every Java program lives in a class. This class is named Main, so the file must be named Main dot java.
- Point to line 2. **Say:** This is the main method. Java starts running here. You'll type this line in every program, so you don't need every word yet.
- Point to lines 4 and 5. **Say:** Each opening brace has a closing brace. Line 4 closes main, and line 5 closes the class. Java reads the braces, not the indents.

Slide 5. Here are two words about `args`.

- Point to line 2 on the slide before. **Say:** The main line ends with String, two brackets and args. That part holds words from the command line.
- **Say:** The command line is where you type a command to run a program. Java Main runs the class Main.
- **Say:** Words typed after the program's name are command-line arguments. Java Main red 7 gives args 2 words, red and 7. Java Main alone gives it none.

Slide 6. Here are four words for today.

- **Say:** A statement is one step, like one print line. A print line ends with a semicolon.
- **Say:** In Python, a new line ends a step. Java doesn't read line breaks, so a missing semicolon is a mistake.
- **Say:** A comment is a note for people. Java skips it. Two slashes start a comment that runs to the end of the line.
- **Say:** A block comment starts with slash star and ends with star slash. Programmers use it to turn off a line while they test.

Slide 7. Predict what this code prints.

- **Say:** Here's Beatrice's code from the bell ringer. Read it from top to bottom, one line at a time.
- **Say:** Write what prints in the What prints box in your notes before we look.
- Walk around and look for students who write three lines, one for each line of code.

Slide 8. `println` ends the line, and `print` doesn't.

- Point to line 1. **Say:** print shows its text and stays on the same line.
- Point to line 2. **Say:** So Beatrice prints right after it. Then println ends the line.
- Point to line 3. **Say:** This text starts a new line. Three lines of code print only two lines. Beatrice was wrong, and that's how we learn.

Slide 9. Practice your code in 1.1 Try It.

- **Say:** Open a browser and type the link on the screen. It opens 1.1 Try It in CodeGrove.
- **Say:** Write the code for each task, and press Check until it works. This is practice, so nothing is turned in.
- **Say:** When I say time is up, close your computer. The exit ticket is on paper.

Slide 10. Big idea: Every Java program lives in a class.

- **Say:** Here's today's big idea. Every Java program lives in a class, and Java starts it in main.
- Hand out the exit ticket.

Day 2 — Java Compiles Before It Runs

Slide 11. Day 2 divider: Java Compiles Before It Runs

- **Say:** Today you will learn that Java compiles the whole file before it runs, and that a compile error names the line where you start looking.

Slide 12. Java names the line it can't finish.

- **Say:** Here's Clayton's code from the bell ringer. On the left, Java could not compile it.
- Point to the red line. **Say:** Java names this line, the one with no semicolon. Check your bell ringer answer.
- Point to the right side. **Say:** With the semicolon back, the program compiles and runs. All three lines print.

Slide 13. One compile error stops the whole program.

- **Say:** Before you look at the Error box, predict. Will line 1 print?
- Point to the Error box. **Say:** The text on line 2 has no closing quote mark, so Java can't tell where the text ends.
- **Say:** Java compiles the whole file before it runs any line. One compile error means nothing runs, so even line 1 doesn't print.

Slide 14. Here are four words for today.

- **Say:** To compile means to check the whole file and turn it into code the computer can run.
- **Say:** The compiler is the tool that does it. When you compile Java, the compiler makes class files full of bytecodes, and the Java Virtual Machine runs them.
- **Say:** A compile error stops the compiler, so nothing runs. The error message names a line. Start there, then check the line just above it.
- **Say:** Run means Java carries out the steps from top to bottom. It happens only after the file compiles.

Slide 15. Fix the Receipt Printer in CodeGrove.

- Hand out the Receipt Printer lab sheet, 1 per student.
- **Say:** Type the CodeGrove link from your sheet. It opens the level, and you need no account.
- **Say:** Fix one error, then compile again. Java may show a new error once the first one is gone.
- Walk around and look for students who stop at the first error. **Ask:** What does the message say now?
- **Say:** After a task passes Check, write the lines the sheet asks for and a Why answer. Then start the next task.

Slide 16. Big idea: Java compiles the whole file before it runs.

- **Say:** Here's today's big idea. Java checks the whole file first, so one compile error stops everything.
- Hand out the exit ticket after students finish their Why answers.

Day 3 — Quiz: Java Program Parts and Compile Errors

Slide 17. Day 3 divider: Quiz: Java Program Parts and Compile Errors

- **Say:** Today you will review the parts of a Java program, `print` and `println`, and compile errors, then show what you know on the quiz.

Slide 18. How many lines does this code print?

- **Say:** Here's Ahmed's code from the bell ringer. Count the lines it prints, not the lines of code.
- **Ask:** What does it print? Have students vote with 1, 2 or 3 fingers.
- **Say:** Hold on to your vote. We'll check it on the next slide.

Slide 19. Only `println` starts a new line.

- Point to line 1. **Say:** print shows Get and stays on the line, so set prints right after it.
- Point to line 2. **Say:** println ends the line. That's why Go starts a new line.
- **Say:** Option 2 is right. To print everything on one line, Ahmed would use print on line 2.

Slide 20. Java cares about capital letters.

- **Say:** The only change is one letter on line 2. System needs a capital S.
- Point to the error box. **Say:** Java reads system with a small s as a different name, so it can't compile the file. Nothing prints, not even line 1.
- Point to the right side. **Say:** With the capital S, both lines print. In the applied task, read each error message, fix it, then compile again.

Slide 21. Name the parts of a Java program.

- Point to line 1. **Say:** Two slashes start a comment. Java skips it, so it never prints.
- Point to lines 2 and 3. **Say:** The class holds the main method, and Java starts running in main.
- Point to lines 5 and 6. **Say:** Each closing brace matches an opening brace. Line 5 closes main, and line 6 closes the class.

Slide 22. Big idea: Java checks the whole file, then runs it from the top.

- **Say:** Here's the lesson's big idea. Java checks the whole file before it runs any line, and then it starts in main.
- Hand out the exit ticket after the quiz.