

Lesson 2: Java Lab

45 minutes

Overview

What are some of the primary components of a Java program?

Students are introduced to the Java Lab programming environment to navigate Java source code files and identify correct Java syntax. Students experiment with the structure of a Java program to identify syntax rules of the Java programming language. While experimenting with Java syntax, students explore The Neighborhood and how the `Painter` can perform a task when given a set of instructions.

Agenda

Warm Up (5 minutes)

Human Computers

Activity (30 minutes)

Adding Students to Your Section

Navigating Java Lab

Wrap Up (10 minutes)

Revisiting the Human Computers

Assessment: Check for Understanding

Objectives

Students will be able to:

- Create a Java source code file with correct structure and syntax
- Identify Java syntax rules
- Identify and correct syntax errors

Preparation

- Create a verified teacher account and a section for your class as indicated on the Code Studio Setup resource
- Print copies of the Unit 1 Guide handout (one for each student)
- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **Java Lab Basics** - Video
- **U1L2 Extra Practice** - Handout
- **Unit 1 Guide** - Handout
- **Writing Our First Program** - Video

Vocabulary

- **class header** - consists of the class keyword and the name of the class

- **integrated development environment (IDE)** - a software application for writing, compiling, testing, and debugging program code
- **software** - a collection of instructions that is run by a computer
- **source code** - a collection of programming commands
- **syntax** - the rules for how a programmer must write code for a computer to understand
- **syntax error** - a mistake in the code that does not follow a programming language's syntax


Teaching Guide


Warm Up (5 minutes)

Human Computers

Remarks

Computing and programming used to have different meanings than they have today. Before electronic computers were invented, a computer was a person who performs mathematical calculations. From the late nineteenth century onwards, people had to perform long, tedious calculations. It was common for women to do this job.

 **Do This:** Share the story of the ENIAC project and the female programmers that operated it.

 **Do This:** Direct students to write down ideas and thoughts in response to the prompt on a sticky note or scrap piece of paper.

Teaching Tip

Students participate in a Give One, Get One activity during the wrap up to share their ideas with their peers.

Throughout the curriculum, students are exposed to examples of computer science in different industries or problems that could potentially be solved using computer science. These are intended to contribute to the development of their identities as software engineers and challenge their perceptions about who software engineers are and what they do. As you learn about your students and their personal interests, you can change these scenarios to show examples of computer science connected to their personal interests to make these more meaningful for your classroom.




Remarks

Be sure to hang on to your notes! We will be using the thoughts you wrote down at the end of the lesson.

Activity (30 minutes)

Adding Students to Your Section (10 minutes)

Do This: Direct students to create a Code Studio account and join your class section:

-  With a personal email account
-  With a Google Classroom account
-  With a Clever account

Teaching Tip

Only show the slide that matches how your class will login to Code Studio.

Code Studio displays students alphabetically by first name. You can choose to have students enter their names as Last Name, First Name so that they are shown alphabetically by last name.

You can confirm that a student successfully joined your section from the Teacher Dashboard. Students should also see a small green bar at the top of their page that says, "You've successfully joined."

 **Do This:** Direct students to access the Code.org home page and navigate to the course.

 **Do This:** Direct students to choose Unit 1 and navigate to Lesson 2.


Teaching Tip

Adding students to your section may take more time if your students do not already have Code Studio accounts. You may need to modify the pacing for this lesson accordingly.

Navigating Java Lab (20 minutes)

Remarks

Software engineers use various tools to write and test programs. In this class, we will use Java Lab as our programming tool. We will explore Java Lab's features and learn how Java files are structured.

 **Do This:** Review the lesson objectives.


 **Distribute:** Give each student a copy of the Unit 1 Guide.

Teaching Tip

The unit guide provides students with guided notes, syntax, and vocabulary for new concepts introduced in each lesson. You provide the entire unit guide, or you can distribute each section at the start of each lesson. You can choose to have students complete the guided notes during the lesson, or you can choose to incorporate the unit guide differently according to the needs of your classroom and students.


 **Do This:** Click through the animated slide to define *integrated development environment (IDE)*.

Remarks

 Java Lab is the IDE we will use in this class. With it, we can write our source code, which will contain our programming commands.

 **Display:** Show the video – *Java Lab Basics*.

Group: Place students in pairs.

 **Do This:** Direct students to Level 1 on Code Studio to complete Levels 1 through 4. Students investigate the programs on Levels 1 and 2 with their partner and make changes to the programs as prompted. On Level 3, students complete a Check for Understanding to identify the error in the program. On Level 4, students write the class header and `main` method.

 1-4

Investigate: Java Basics





 Teaching Tip

Encourage students to focus on understanding how Java code is written rather than the functionality of the programs. Emphasize that they will learn about how these code works throughout the unit.

Remarks

Now that you have had a chance to get comfortable with Java Lab and look at a Java file, let's review the key components and features you found.

 **Display:** Show the video – *Writing Our First Program*.

 **Do This:** Explain the Java syntax rules.

 **Do This:** Click through the animated slide to identify the components of a class.

Remarks

On Level 3, you used your new knowledge of Java syntax to find the error in the program. Software engineers spend a lot of time finding and fixing errors in programs. The error that you encountered is called a syntax error.

 **Discuss:** Use the Retrieve-Pair-Share strategy to discuss the prompt.

- *Now that you have learned about syntax, what do you think it means to have a syntax error?*

Discussion Goal: Students make predictions about the definition of syntax error based on what they know about syntax. Students may share examples that might cause a syntax error, such as forgetting a semicolon or writing the main method incorrectly.

 Teaching Tip

The Retrieve-Pair-Share strategy facilitates this discussion by directing students to write down their thoughts and ideas first, then pair up students to compare their answers and expand their notes. Have students share the ideas they discussed with their partners as a class.

 **Do This:** Define *syntax error*.

 **Do This:** Click through the animated slide to identify example syntax errors.

 Teaching Tip

Point out the error in the code segment after showing the result in the console:

- For the "your program does not contain a main method" error, have students look at the `main` method to identify the missing `[]` after `String`.
- For the "class Myconsole is public" error, have students look at the name of the class on Line 1 and the name of the file to identify the difference in capitalization (lowercase c in the code versus uppercase C in the file name).


Share an example of syntax errors you experienced as you were learning Java.

Wrap Up (10 minutes)

Revisiting the Human Computers

Remarks

At the beginning of this lesson, we heard about the story of the ENIAC project and the female programmers that operated it. Let's revisit the ideas and thoughts you had and share these with your peers. It's time to...

 **Do This:** Play the music clip to cue the Give One, Get One activity and direct students to participate in a Give One, Get One.

Teaching Tip

Using music to cue transitions helps create routines in the classroom. When students hear cues, they know it is time to transition to a specific activity. Emphasize to students what the music clip means, and be consistent with the use of the music clip to cue the activity.

Click the blue play icon on the slide to play the music cue.

Give One, Get One encourages students to seek and share ideas and information. Students circulate the classroom and pair up with a partner when prompted by the teacher. Each student "gives" or shares one of their ideas as the other student "gets" or listens and writes it down. The teacher then prompts students to find new partners and repeat the process.

 **Discuss:** Click through the animated slide to display the prompts.

- *What did you learn from each other?*
- *How do these ideas and thoughts contribute to your identity as a software engineer?*

Discussion Goal: Students share the ideas they gathered and identify patterns or common views. Students recognize the similarities and differences between being a programmer on the ENIAC and using Java Lab and make personal connections to their identity as software engineers.

 **Do This:** Review the concepts covered in this lesson.

 **Display:** Key Vocabulary

Assessment: Check for Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. You

can use these questions as an exit ticket.

☰ 5



Check for Understanding



This curriculum is available under a
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes **contact us**.