Lesson 11: Debugging Strategies

45 minutes

Overview

How can the console be used as a debugging tool?

Students learn to print information to the console as a debugging tool to identify logic errors in their programs. Students expand an existing subclass to implement a new method that displays information about the state of an object and use this method within other methods to troubleshoot errors.

Standards

Full Course Alignment

CSA Conceptual Framework

- ▶ **MOD-1** - Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence
- ▶ **VAR-1** - To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values

Agenda

Warm Up (5 minutes)
Finding Errors

Activity (30 minutes)
Debugging Programs
Getting Information About the Painter

Wrap Up (10 minutes)
Debugging Wall
Assessment: Check for Understanding
AP Classroom Topic Questions

Objectives

Students will be able to:

- Debug common errors in code
- Use `System.out.print()` and `System.out.println()` to output information about the state of an object to the console
- Write a `void` method to add a new behavior to an existing class

Preparation

- 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

- **Printing in Java** - Video
- **U1L11 Extra Practice** - Handout

Vocabulary

- **concatenation** - joining two strings together

Teaching Guide

Warm Up (5 minutes)

Finding Errors

Remarks

> We have gotten pretty good at finding and fixing syntax and logic errors! Logic errors tend to be tricky to find, but there are strategies and tools we can use to help.

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *When have you encountered logic errors in your programs?*
- *How did you find and fix these errors?*
- *What tools or strategies would help find and fix logic errors?*

Discussion Goal: Students share examples of logic errors they have encountered in their programs and their strategies to find and fix these errors. Students suggest tools or strategies for finding logic errors, such as stepping through their code one line at a time or seeing the current state of a `PainterPlus` object as it navigates and paints The Neighborhood.

Activity (30 minutes)

Debugging Programs (10 minutes)

Remarks

> Logic errors can be tricky to find. However, software engineers use the console to print information to help them find these types of errors. Let's find out how we can do that.

Do This: Review the lesson objectives.

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *What do these methods return?*
- *What do you think these methods do?*
- *Why would this information be useful?*

Discussion Goal: Students suggest that the method return the x and y coordinates, the direction, and the amount of paint for a `PainterPlus` object and realize that methods can return more than just `boolean` values. Students note that these methods provide information about the current state of a `PainterPlus` object and share ideas for when this information would be useful, including debugging logic errors.

Teaching Tip

Recall good programming style by asking students how they knew what these methods do and if they could have determined their purpose if they were named differently.

Display: Show the video – *Printing in Java*.

Do This: Click through the animated slide to discuss the example code segment.

Getting Information About the Painter (20 minutes)

Remarks

> The `Painter` class gives us some of the basic behaviors we need to navigate and paint The Neighborhood. We can add a method to our `PainterPlus` class to print information about a `PainterPlus` object to track its state.

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *What would the method signature be for this new method?*
- *What would we write in the body of this method?*

Discussion Goal: Students identify the method as a `void` method and share ideas for the method name. Students suggest printing the values of the x and y coordinates, the direction, and the amount of paint in the method's body.

Teaching Tip

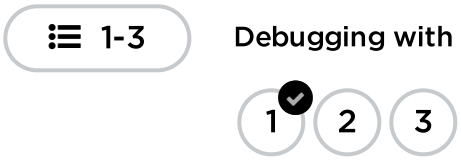
Have students consider logic errors they have encountered before. Ask students what would have been helpful to know as the `Painter` was moving and painting in The Neighborhood.

Do This: Direct students to add the new method to the `PainterPlus` UML diagram.

Remarks

> Let's add this method to our `PainterPlus` class!

Do This: Direct students to Level 1 on Code Studio to complete Levels 1, 2, and 3. Students complete a Check for Understanding on Level 1, then continue to Level 2 to write a method to print information about the state of a `PainterPlus` object. Students use the new method while solving a problem on Level 3.



Remarks

> This is a good time to commit our code and save our new version of the `PainterPlus` class to the Backpack. Anytime we make changes to our programs, it is helpful to commit, or save, our work as a new version in case we need to revert to a previous version.

Do This: Play the music clip to cue committing their code and saving the new version of the `PainterPlus` class to the Backpack.

Wrap Up (10 minutes)

Debugging Wall

Remarks

> We saw some new types of errors today! Let's discuss these errors we encountered and update our

> Debugging Wall with the strategies we used to debug them.

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *What errors did you encounter while writing your code?*
- *How did you find and fix the error?*

Discussion Goal: Students consider errors they encountered in previous lessons and this lesson and share scenarios. Students suggest using their new method to find and fix logic errors they have encountered.

Do This: Have students choose a strategy as a class and add it to the Debugging Wall.

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.

Check for Understanding

AP Classroom Topic Questions

To assign questions from the AP Classroom Question Bank that align with this lesson, create a custom quiz in AP Classroom by searching the Question Bank for the Essential Knowledge statements listed at the top of this lesson plan. You can find instructions and video demonstrations to do this on **AP Central**.

The following Topic Questions in AP Classroom can be assigned as a formative assessment for this lesson:

- Topic Questions 1.1

Note: *Some Learning Objectives and Essential Knowledge statements in the suggested Topic Questions are covered in later units.*