Java

Lesson: How will playing "debugger" help us understand the concepts of debugging code?

Do Now (7mins)

In your journals, write what you believe the definition of debugging is. If you do not know or have not heard the term, write an educated guess.

Do Now review

Objectives

- review of methods by debugging a sample code
- understand different ways to debug a problem
- come up with strategies to solve a problem
- rely on their peers to find solutions to problems

What is debugging?

Debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.

The terms "bug" and "debugging" are popularly attributed to Admiral Grace Hopper in the 1940s. While she was working on a Mark II computer at Harvard University, her associates discovered a moth stuck in a relay and thereby impeding operation, whereupon she remarked that they were "debugging" the system. However, the term "bug", in the sense of "technical error", dates back at least to 1878 and Thomas Edison (see software bug for a full discussion).

Let’s play Debugger!

This is an unplugged activity so please close laptops for now.

Game Description

There is a bug in the classroom that is causing students to malfunction! Two students are chosen to be debuggers to find out how to fix the bug in the system.

Set Up

Two debuggers are given a notepad or dry erase board to take notes. First they choose a difficulty level and then they are sent away to come up with strategies to find the bug in the classroom. Some strategies include:

- Each asking the same yes or no questions
- Coming back to a questions that has weird answers
- Ask questions with obvious answers (is the sky blue?/ what color is the sky?)

Everyone else agrees on what the bug in the classroom will be. Some common examples are:

- People with a red shirt will lie
- If your name starts with the letter D, clap before you answer the questions
- If debugger A asks a question, say "I don't know" but always answer debugger B's questions
- If students are grouped by tables, students at one table can say, "I dont know, ask table X."

Gameplay

The 2 debuggers come back (after 2 mins or so) and begin their investigations. They can ask anyone any question. They will have a set amount of time(6-8 mins) to find the bug before the classroom is irreparable. If a student answers in a way that violates the rules of the bug, either the teacher or the students in the classroom can cross their arms and say "Error!" That student can try answering the question again.

Debuggers are encouraged to take notes on their findings throughout the game.

The game ends when time is up or when the 2 debuggers find the bug. They will have a chance once time is up to guess as well.

Let’s play!

Debuggers: take notes, ask questions, you have 6-8 mins!

Everyone else (but especially bugs): don’t forget the rule we chose!

Whole-group Debrief

How do you think this game relates to debugging a coding problem?
What strategies translate or don't translate well?

Some strategies that can be used in the game and in code debugging:

1. Looking in different areas of your code.
2. Asking obvious questions like "Did I miss a parenthesis? Am I missing a semicolon?"
3. Working with a partner to find the bug.
4. Asking clarifying questions about the rules of coding.
5. Working through the problem and giving yourself a set amount of time before asking for help.
6. Sometimes the answer was right under your nose the entire time!

(and anything else that comes in the group discussion)

Student Activity (10 mins)

You have been given a sample code with 5 bugs in it (who knows, there may be more!). Work with a partner to follow the instructions in the lab and find the 5 bugs. Try to find and fix at least 3!

Review (5mins)