

Lesson 14: Find that Bug!

Powerful Ideas of Computer Science	Debugging, Design Process
Powerful Ideas of Literacy	Writing Process, Editing and Audience Awareness, Phonological Awareness, Alphabet and Letter-Sound Correspondence
PTD	Communication
Palette of Virtues	Perseverance, Forgiveness
Children will be able to...	● Define and celebrate the process of debugging. ● Identify strategies for debugging and editing. ● Locate errors or “bugs” in ScratchJr code. ● Locate errors in English text (e.g., spelling errors, letters written incorrectly).
Vocabulary	● Debugging: to find problems in a computer program and to try to solve the problem in different ways
Teacher Preparation	● Read lesson plan. ● Have a tab open for the Anchor Chart for Design Process slideshow ready to be projected. ● Teacher Project to be Debugged: Create your own buggy project or recreate the screenshot below.
Warm Up ● Design Process Song (<i>Suggested Time: 5 minutes</i>) ○ Remind the children of the <i>Design Process</i> song and sing it again as a class! ○ Display the Design Process Anchor Chart. <i>(to the tune of “Twinkle, Twinkle”)</i> <u>Ask and imagine, plan and create.</u> <u>Test and improve and share what we make.</u> <u>(Repeat)</u> Opening Tech Circle ● Test and Improve (<i>Suggested Time: 5 minutes</i>) ○ Show the Design Process chart again and remind children of the steps for making a ScratchJr Project: Ask, Imagine, Plan, Create, Test and Improve, Share! Today we	

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) - K © [2020 - 2025] DevTech Research Group. Some Rights Reserved.

Coding As Another Language (CAL) for ScratchJr (CAL-ScratchJr) - K is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>

This license requires that reusers give credit to the creator. You may distribute, remix, adapt, and build upon the material in any medium or format, but must license the modified material under identical terms and indicate what has changed from the original. You may not use or adapt this work for commercial purposes.

will be learning about how to Test and Improve!

- Tell children that we need to test our programs to see if they follow our plan, and if they don't... we need to improve!
- **Debugging** (*Suggested Time: 10 minutes*)
 - When programs don't follow the plan, we call that a BUG, and improving is called debugging.
 - **Debugging** is a word used in computer science to describe when people find problems in their computer programs and try to solve the problem in different ways. Discuss with children about how to encounter problems. Programmers go through the iterative process of debugging to find and fix problems.
 - Remind children of how Katherine went over all the numbers again to fix bugs, and that we can do the same with our work!

Unplugged Time

- **Debug the Teacher** (*Suggested Time: 10 minutes*)
 - Children give instructions to “program” the teacher to have the teacher do something.
 - E.g., How to draw a house.
 - When children give instructions, they'll need to be very specific for the teacher to complete the program.
 - If instructions are not clear, do it “wrong” based on instructions (e.g., if children say “draw a square,” teacher just makes a shape in the air until children say, “get a pen and paper”).
 - Children will have to debug their instructions when they are wrong.

ScratchJr Time

Structure Challenge:

- **ScratchJr Problems** (*Suggested Time: 10 minutes*)
 1. Pull up your buggy project and explain to children what your plan was (e.g., what you want it to do). Then show the children the project and ask them to think (not to share yet) of possible bugs and how they would debug the program. Recreate this project in ScratchJr.



2. Explain to the children that your plan for Cat was to move to the right twice and get smaller.
3. Play your buggy project for the children.
4. Have children take turns raising their hand and **debugging** things in your project. Make sure to stop and play the program after each change. Tell children that you are **testing**. Remind them that that was part of our Design Process!

Closing Tech Circle

- **Group Reflection** (*Suggested Time: 5 minutes*)
 - Have children think about a time that they made something that didn't turn out like they planned! Have each child share their story and one thing they did, or would do, to **improve/debug** what they made!
 - This is a good opportunity to discuss the importance of perseverance and forgiveness during the debugging process (e.g., remind children that it's okay to make mistakes -- reintroduce "I am a CODER" mnemonic to help students through frustrating moments of the coding process).