



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #5: Debugging



Essential Question

- Do you need to be creative to be good at computer programming?

Lesson Overview

In Lesson 5, students will dive deeper into one key component of the Programming Design Process: debugging! By using trial and error to correct code in this week's Pencil Code Cards, students will identify bugs and fix them. Being able to debug a program is a skill that will help the students be better able to make their innovative ideas a reality. Additionally, students will continue to work on projects and programs that inspire them and prepare to compile their WOW! Portfolios.

Lesson Objectives

Standard #2: Citizen Schools students will use a design process to create ideas or products.

Lesson Objectives

- SWBAT test and revise their programming language to make at least two Pencil Code Cards with simple coding errors function properly.
- SWBAT become familiar with three ways professionals use programming in their careers.

Standard #4: Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.

Lesson Objectives

- SWBAT find and correct "bugs" on at least two Pencil Code Cards.

Lesson Agenda

5 Minutes	Hook: Feature Project
10 Minutes	Introduction to New Material: Debugging
25 Minutes	Activity 1: Cards
15 Minutes	Activity 2: Game OR Share Projects
30 Minutes	Activity 3: Careers in Programming
5 Minutes	Assessment

Lesson Preparation

- Preparation:
 - Practice the Pencil Code Cards for this week. Samples of all of the Pencil Code Cards are available at citizenschools.pencilcode.net. Username: citizenschools, Password: Citizen. Each card is saved in accordance with the file-saving procedure taught in this apprenticeship (week#/cardname). Innovation ideas are included for some cards.
 - Prepare words to add to the Word Wall.
 - Prepare copies of handouts.
 - Review student assessment work from the previous week. Write a note to the student appreciating some aspect of their work, making a recommendation for further innovation, and recognizing their



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #5: Debugging



work and learning.

- Pick a project from Week 4 to feature for the Hook. Ideally you feature a different student's work each week. Select a student whose work is exemplary and might push other students' innovations, thinking, and creativity further.
 - Prepare to show students how to debug programs in Pencil Code. Plan to use a teacher computer connected to a projector to demonstrate. Minimal direct teach is needed this week as the Pencil Code Cards provide hints that are sufficient.
 - Prepare a simple program to debug and use as a model in Activity 1. Alternatively, use Pencil Code Card: Debug: Spiral. If your students have been completing 100% of the cards, you may want to create a program to use as demonstration. If, however, your students love to dive deep on only a few cards, then using one to demonstrate won't matter.
 - Watch the [video](#) "What Most Schools Don't Teach" by code.org. Cue it up ahead of class to play during Activity 3.
 - Coordinate a panel of guest speakers to speak during Activity 3 about their professional experiences in computer programming. See the Unit Guide for more resources and find a sample letter to panelists in the Lesson Resources. Depending on how much time you need for your guest speakers, you may want to eliminate or shorten Activity 2.
 - Plan ahead to provide a tangible thank you to your guests. Perhaps a thank you note, Citizen Schools swag, or a certificate of appreciation.
- Co-teaching plan:
 - If you have a co-teacher, consider assigning him or her to "guest" duty. Greet the guest speakers, welcome them to the school, show them where the restroom and the classroom are, etc.
 - If you have multiple co-teachers, you could split your panel up into a carousel rotation. Instead of having the panel speak to the whole group of students, split your speakers up and divide your class into the same number of groups as speakers. Have each group of students spend a certain amount of time with one speaker before rotating to the next speaker. This will promote a more informal discussion atmosphere and give students more individual time with each guest.

Materials

Every Class:

- Computer with internet access (one per student)
- Computer with projector (one per class)
- Student folders (one per student)
 - Student Guide*
 - Pencil Code Cards completed by each student (from previous weeks)
- Previous weeks' cards organized in a binder or file box
- Roadmap to WOW!
- Expectations poster
- Word Wall
- Stickers (to indicate earning badges)
- Programming Design Process visual

This Class:

- Week 5 Pencil Code Cards (one per student per card)*
 - (Debug: Spiral, Turtle Pen, Bullseye, House, Cat, Beads)
- Word Wall words and definitions (Bugs, Debugging, Remix)
- Handout - Innovation - Lesson #5 (one per student)
- Handout - Careers in Programming - Lesson #5 (one per student)



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #5: Debugging



- Consider adding name tags or table tents with names of guest speakers
- Thank you item for guest speakers (one per speaker)

*These materials are located in the lesson folder, not in the Lesson Resources at the end of this lesson.

**Hook:
Feature Project
(5 Minutes)**

Teacher's Note: If students are no longer excited to see each other's work in this context, consider switching to featuring a more complex program. Begin by showing the program in action, and then showing the code behind the program. Highlight coding concepts that they have learned to show how what they are learning is building towards being able to create programs similar to the one you feature.

- **Say:** Welcome back to Pencil Code! Let's begin this week by taking a look at some of the awesome work you completed last week!
 - **Play** the program from the featured student of the week.
- **Point out** why you selected this project to share. Highlight an innovation, a unique piece of code, a way the student was thinking creatively, or a step in the design process that the student used to successfully problem solve.
- **Ask** the student who is featured if he or she wants to share anything about the project or ask for any feedback.
 - **For example,** the student could share a future innovation that he or she thought would be cool, but isn't sure how to do, or the student could ask for other students ideas of what to try next.
- **Transition:** Today we're going to learn how to find and fix bugs (or trouble spots) in our code. Let's get started!

**Introduction to New Material:
Debug and Remix
(10 Minutes)**

- Briefly **review** concepts (creating functions) covered in Week 4.
- **Review** the Programming Design Process also covered in Week 4.
 - Step 1: We have selected an idea we want to create.
 - Step 2: Write code to make the idea come to life.
 - Step 3: Press "play" and see what happens. If it works to create the image in our minds, great! Done! If it doesn't, continue to Step 4.
 - Step 4: Ask ourselves what went wrong. Is there an error in the code language (a missing comma, an extra parenthesis, etc.)? Is there a direction missing (a right turn, a backwards movement, etc.)? Or is there something that I don't know how to code (animation, multiple turtles, etc.)? Based on the answer, pick Step 5.
 - Step 5a: Check all the details in the code!
 - Step 5b: Add all missing directions!
 - Step 5c: Ask for help!
 - Step 6: Repeat steps 2-5 until the program works just like you imagined it to.
 - Step 7: Celebrate!



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #5: Debugging



- **Say:** Today we're going to dive deeper into this process and think more about steps 4 and 5. I'd love to tell you that once you master programming, then you usually just write perfect code the first try and use Steps 1-3 and then skip to 7, but the reality is that even expert computer programmers--masters of coding--still are confronted with bugs all the time that must be found and fixed in steps 4-6. Today we're going to practice finding bugs and fixing them.
 - **Define:** Bug, Debugging, and Remix
 - **Word Wall Definition: Bugs** = Errors in the computer programming that cause undesired behavior.
 - **Word Wall Definition: Debug** = The process of determining the origin and solution for a bug.
 - **Word Wall Definition: Remix** = Putting together several creations originally made by other people to create something new. Also called a "mashup."
 - **Teach** tips to think about when reviewing code to look for a bug. **Write** these on the board or on a piece of chart paper to save for future reference.
 - Think about what the code is supposed to do and compare that to what the program is actually doing. Use this gap to hone in on the area of the code that needs tweaking.
 - Play "Guess my Number." If the box is too small, pick a larger number. If it's too big, pick a smaller number. Hone in on the right number through trial and error.
 - Look for switched numbers and spelling errors
 - Look for missing indentations, commas, or parentheses
 - Use your resources (Student Guide reference sheets, past cards, teachers and classmates)
- **Review** the basic agenda for the day. Note that today's agenda is different from other classes. Instead of having two coding sessions, we will only have one. In Activity 3 we will be watching a video and welcoming some experts from the field to talk to us more about their careers in computer programming.
 - Agenda:
 - Work time
 - Pause for sharing projects
 - Save an assessment for the week
 - Guest speakers
- **Transition:** Cue the procedure to get folders, turn on computers, and get to work!

Activity One: Cards (30 Minutes)

Teacher's Note: Use your knowledge of a student's interests and abilities to determine which card they should start with. Alternatively, let the students decide for themselves! If you decide to let them decide, assign a process and order by which each student will select a card or hand out all the cards at once and allow them to pick one to get to work.

- **Hand out:** Handout - Innovation - Lesson #5.
 - **Say:** Walk the students through how to complete this handout as they work on the Pencil Code Cards today. Model your thought process as you model debugging either the Debug: Spiral card or a similar program that you create. Use an Elmo to project your own work on a sample Innovation - Lesson #5 handout.
 - **Model** writing down changes you will make to try to fix the problem. Model trying them in the code. Record what happens in each iteration. Be sure to show at least three attempts before you correctly debug the code.
 - **Explain** that students are to do the same thing when debugging two of the cards today. They can work on more than two cards, but they only need to record their attempts to debug on any two cards they choose.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #5: Debugging



- **Hand out cards:** Start each student with any of the Week 5 cards.
- **Say:** Your task is to first start with the code on the card and then compare what the computer does with what you want it to do. Find the bugs and fix the code using the Programming Design Process here (point to poster)!
- **Give** students 25 minutes of work time.
 - Once students complete the Innovation handout for Lesson #5, they may continue to work on Week 5 cards or they can go back to past projects that they are working on.
 - **Remind** students to NAME and SAVE their projects regularly. If needed, set the timer to cue you and the students to remember to save their work.
 - During this work time, CTs should be rotating around the room, checking in with students, assisting as needed, and encouraging students to try a new card or a new step in their programming.
 - Encourage students to switch back and forth between text code and block code as needed.
 - Check in on the Handout - Innovation - Lesson #5 also. Ensure that students have debugged two cards and logged several attempts at debugging for each.
 - CTs should also hand out stickers to award badges for the day.
- **Bring** the group back together and **wrap up** computer time for today.
 - **Say:** Today I will collect the Innovation handouts, along with your submitted project for your assessment. Be sure to pick a project that you are most proud of debugging today. Pick that project now and save it as "week5/assessment." In your student folder, fill out the Week 5 Assessment section.
 - **Walk students through** how to fill out the Week 5 Assessment written portion, if necessary. (See Student Guide).
 - I'm proud of this program because _____
 - Which of the following cards did you successfully debug today?
 - **Say:** We're doing this early today because we are having a guest speaker later in this apprenticeship today. We will still wrap up at the end of class, but we won't be getting back on our computers.
 - **Award badges** for today. There aren't any specific badges aligned with Lesson 5 concepts, but students will likely have completed all the challenges in Badges #1-4.
- **Ask** students to follow the procedure for shutting down the computers.

Activity Two: Share Projects (15 Minutes)

Teacher's Note: Use this time to either share projects that students are working on or share a project that pushes their skills further.

- **Say:** let's take a look at xx student's project this afternoon...
 - **Share** two-four different students' projects. Highlight how students used the Programming Design Process and caught various bugs in their programs. Discuss the difference between a bug that causes an unwanted behavior and something that you did not mean to code, but end up keeping because you like the result. Highlight a few strategies that instructors saw students doing to solve problems in the design process. For example, I wanted to share this project because last week, this student tried these two strategies and that led to solving this challenge or bug.
 - **Share** feedback and ideas for further work.

Alternatively, if you'd rather play a game than share student work, consider playing a quick "Guess the Code" game.



Core Apprenticeship Library

Apprenticeship Sector: Coding & Gaming

Unit Guide: Pencil Code

Lesson #5: Debugging



- **Say:** I'm going to show you sets of written code and the same number of programs being run. The game is to guess which code goes with which program. These are a little trickier than last week! These all feature buttons and key downs!
- **Use** the following links to one debugging challenge and a "Guess My Number" game
 - Debug this [tree program](#)
 - Play [Guess my number](#)
- **Connect to Lesson 4** by highlighting the use of functions in the "Guess My Number" game.
 - See all games available for this section at <http://inventgame.pencilcode.net>

Activity Three: Careers in Programming (30 Minutes)

Teacher's Note: This activity is written for a guest speaker or panel of guest speakers to share their experiences in a career related to computer programming. Depending on how much time you need for your guest speakers, you may want to eliminate or shorten Activity 2. See the Unit Guide for more resources and find a sample letter to panelists in the Lesson Resources.

Teacher's Note: If you have multiple co-teachers, you could split your panel up into a carousel rotation. Instead of having the panel speak to the whole group of students, split your speakers up and divide your class into the same number of groups as speakers. Have each group of students spend a certain amount of time with one speaker before rotating to the next speaker. This will promote a more informal discussion atmosphere and give students more individual time with each guest.

- **Say:** We've talked about careers in programming since day one of this apprenticeship, but now that you really know what programming is, I think it makes sense to dive a little deeper by meeting three people who do this work in very different ways.
 - **Say:** You'll have a chance to ask them questions throughout the panel. Let's talk for a minute about what questions you might want to ask them!
 - **Handout:** Careers in Programming. Walk the students through the sample questions and give them two-three minutes to star questions they want to ask and add any questions of their own.
 - **Explain** that students are to use this handout to take notes during the panel. While you won't be assessing the handout, they are going to put it in their Student Folders and you will take a quick look.
- **Welcome** the speakers.
- **Say:** First, we're going to watch a video that inspired me to teach this apprenticeship.
 - **Play** the [video](#) "What Most Schools Don't Teach" by code.org.
 - **Debrief:**
 - What types of careers use computer science or computer programming?
 - Was anything inspiring to you about this video?
 - What do you want to know more about?
 - What questions do you have after watching it?
- **Introduce** the guest speakers. Share a little bit about what they do and how they use programming in their lines of work.
 - **Ask** each speaker to share a little bit about how they got into computer science and what they love about programming.
 - **Note:** if students are eager to ask questions or if speakers are eager to share specific information, allow the conversation to progress naturally. If, however, students are quiet and speakers are waiting to be prompted, then go ahead and either ask a question yourself or prompt a student to ask one of the questions they starred on the handout.
 - **Points to highlight:**



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #5: Debugging



- Computers cross cut every career
- Coding is becoming a more and more necessary skill
- Programming is a creative field that helps other people
- Coders and programmers need to be able to share their work with potential employers and clients via a portfolio. Employers look for both high-quality work and creativity.
- **Connect:** Make the connection to the WOW! Portfolio that each student will create.
- **Say:** Thank the speakers for their time.
 - **Note:** Plan ahead to provide a tangible thank you to your guests. Perhaps give a thank you note, Citizen Schools swag, or a certificate of appreciation.
- **Transition** to the Assessment.

**Assessment
Work Product
(5 Minutes)**

- **Say:** Today I will collect the Innovation - Lesson #5 handouts along with your submitted project for your assessment. You have already picked that project and saved it!
 - **Say:** Before next week I will check out your work and next week I'll give you some feedback!
 - **Note:** If there is time available, share one or two student projects now.
- **Connect to WOW!** Taking this apprenticeship could be the first step for you on a path to a career in computer programming. Your WOW! Portfolios will mimic a portfolio from a new college graduate, applying for his/ her first computer programming position.
- **Connect to Week 6:** Next week we will learn how to add both images AND animation to our programs!
- **Ensure** all students have received stickers for the badges they have earned!



Word Wall - Lesson #5

Bugs

Errors in the computer programming that cause undesired behavior.

Debug

The process of determining the origin and solution for a bug.

Remix

Putting together several creations originally made by other people to create something new. Also called a “mashup.”



Handout - Innovation - Lesson #5

Name: _____

Card #1: _____

Draw a picture of...

what it's supposed to do:

&

what it's actually doing:

--	--

List changes that you made to the code in
an attempt to debug it.

What happened?
(Draw or write what happened)

1. _____

—

—

2. _____

—

—

3. _____

—

—



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #5: Debugging



4. _____

—

—

5. _____

—

—

Card #2: _____

Draw a picture of...

what it's supposed to do:

&

what it's actually doing:

--	--

List changes that you made to the code in
an attempt to debug it.

What happened?
(Draw or write what happened)

1. _____

—

—



Core Apprenticeship Library
Apprenticeship Sector: Coding & Gaming
Unit Guide: Pencil Code
Lesson #5: Debugging



2. _____

—

—

3. _____

—

—

4. _____

—

—

5. _____

—

—



Handout - Careers in Programming - Lesson #5

Name: _____

Questions for Speakers:

- How did you get into computer programming?
- What makes a great programmer?
- When you are hiring a new programmer, what do you look for?
- Why should I consider programming as a career?
- What did you want to be when you were my age?
- How much can I expect to earn as a programmer?
- How long will I need to go to school to be a programmer?
- What degree should I get in college?
- What classes should I take in high school?

- _____
- _____
- _____
- _____
- _____

Notes on Video:

Notes on Panel Presentation:



Sample Email to Guest Speakers

Dear Guest Speakers,

Thank you so much for volunteering to spend 30-60 minutes with our students next week during our apprenticeship! The goal of your visit is to expand the students' understanding of what it means to be a computer programmer and inspire them to pursue a career in programming. To support these efforts, please plan to give a brief overview of your background, how/ why you became a computer programmer, and what your day-to-day work is like. Additionally, please share why you like doing this work.

The panel will take place during the last 30 minutes of our class. I'd like it to feel like an informal Q&A session, so don't feel like you need to prepare a presentation, although it would be helpful to think through key points you want to make beforehand. The students will also be prepared to ask you a few questions. Some sample questions they might ask are:

- What makes a great programmer?
- What did you want to be when you were my age?
- How much can I expect to earn as a programmer?
- What classes should I take in high school?

Thank you so much again for making time to come visit. We are looking forward to it! Please don't hesitate to contact me directly if you have any questions or concerns. You can reach me at this email address or at <insert phone number> via call or text.

Thanks,

<insert name>