

Code Your Moves

Unplugged coding: Use pseudocode to write a program for a dance move.

🕒 45 min

📖 Beginner

🎓 Grades 6–8

💻 Hybrid



Teacher Support

Key objectives

Students will:

- Create and follow a set of step-by-step pseudocode instructions
- Decompose, identify, and fix problems in their pseudocode

Things you will need

- Lesson Presentation (see Additional Resources)
- Paper (2 pieces per student)
- Writing/drawing tools (pens, pencils, markers)

Additional resources

[Lesson Presentation](#)

Hybrid learning resources

[5 Ideas and Examples for Blended Learning Using the 5E Model](#)

Overview

- This is an unplugged coding lesson designed for hybrid learning. You don't need a LEGO® Education SPIKE™ Prime Set, LEGO bricks, or software to teach this lesson.
- In this lesson, your students will write pseudocode that tells how to perform a dance move. They'll share their pseudocode with a partner.

You can teach this lesson:

- As an introductory lesson to kick off the Life Hacks unit
- Before or after the *Break Dance* lesson, to introduce or reinforce how to write pseudocode
- Before *The Coach* lesson to help lay the foundation for the open-ended project at the end of the unit to extend your students' learning

- A visual Lesson Presentation is provided in the Additional Resources section of this lesson, but this presentation is not required to teach this lesson.

this lesson plan. Use this presentation to pace your lesson and guide your students through each of the 5 Es.

Coding Concepts

- **Algorithm:** A step-by-step set of instructions for completing a task.
 - **Bug:** A coding error. Debugging is the process of fixing the error.
 - **Decomposition:** Breaking down code into smaller parts or smaller problems.
 - **Pseudocode:** A written plan outlining the steps in an algorithm before it's coded on a computer. Pseudocode helps programmers identify and anticipate problems (bugs), and plan for decomposing code steps.
-

Engage

(10 Minutes)

Facilitate a quick discussion to uncover your students' background knowledge of code.

- Through this discussion, you and your students will develop shared definitions of code and coding.

Ask questions like these:

- What is code? *(Sample responses: something that's used to tell a machine or robot to accomplish a task; something fun or creative like programming a game on a computer.)*
- Have you heard of any other words for code? *(Some students who are familiar with coding might share the words, "algorithm, pseudocode, and bugs." Have students who are familiar with these concepts share their knowledge.)*
- What does code look like? *(Sample responses: blocks, text, numbers)*
- Explain that code is a set of directions.

Display the first *Engage* page of the Lesson Presentation.

- This page shows examples of a hand-clapping game.

Continue the conversation by asking questions like these:

- What's happening in this picture? *(Sample response: They're playing a hand-clapping game.)*
- What kinds of hand-clapping games have you played? *(Have one or two of your students share their experiences.)*
- How would you explain to someone how to play that game? *(Invite one or two of your students to share their ideas.)*
- What would an ideal set of directions look like? *(Encourage your students to help define the criteria for an ideal set of directions. For example, "clear, organized, and in the right order")*

and in the right order.)

- Tell your students that by giving directions for how to play a game, they've just practiced making an algorithm. Explain that an algorithm is a step-by-step set of instructions for completing a task.

Display the second *Engage* page of the Lesson Presentation.

- This page shows examples of written sets of directions or "code." Choreographers use visual notation to document dance steps, coaches use sketches and words to document sports plays, and sheet music contains musical notes to direct musicians on what and how to play.

Continue the discussion with questions like these to help your students make connections between real-life non-computer "code" and the pseudocode they'll be writing later in this lesson:

- What do you notice about the choreographer's notations? (*Sample response: The four boxes show you the order of the movements, the arrow shows how to move, steps 1 and 4 are repeated.*)
- What do you notice about the coach's sketch? (*Sample responses: the players are represented by Xs and circles; there are arrows to indicate the movement*)
- What do you notice about the musical notes? (*Sample responses: the song is "Twinkle Twinkle Little Star;" the circles represent whole notes*)
- What do these directions/algorithms have in common? (*Prompt your students to build on the criteria they've defined above for an ideal set of directions.*)
- How do these examples of written directions help the choreographer, coach, and musician? (*Prompt your students to discover that these directions help them with planning and to discover potential errors.*)

Display the third *Engage* page of the Lesson Presentation.

- This page shows an example of SPIKE Prime code from the Break Dance lesson, which is part of the Life Hacks unit. This programming stack controls the SPIKE Prime Break Dancer.

Play the video at the top of this lesson plan page to show your students the Break Dancer robot in action. Then use talking points like these to continue the discussion:

- This is the programming stack for the Break Dancer robot.
- What's the code telling the robot to do? (The steps are written as pseudocode below. To add a playful element to the lesson, invite students to role-play the robot's movements.)
- What would you call this dance move? (Encourage your students to come up with names for the dance move "just for fun" and to prepare them for naming the dance move they'll write pseudocode for later on.)
- How would you document this code on paper as an algorithm? (Have one or two of your students share examples of how they'd document the code using words or pictures.)
- Tell your students that by explaining the code in their own words, they've just practiced creating an algorithm and writing pseudocode

Sample Pseudocode Solution

Your students might share different ideas for explaining this programming stack using words. This is one example:

1. Program receives Go!
2. Motors D and F run that way for 1 rotation
3. Motors wait 1 second
4. Motors repeat the movements 10 times

Hybrid Learning Tips for the *Engage* Phase

Synchronous

- Facilitate a whole-class virtual discussion during your scheduled lesson time. Keep your talking to a minimum. Encourage your students to answer or pose questions of their own. Try using an online polling tool or chat function to get everyone participating.

Asynchronous

- Display the *Engage* pages of the Lesson Presentation, and choose whether your students should respond directly to you individually or collectively on a class discussion board. Online discussion boards are a great way to encourage peer-to-peer engagement without requiring all students to be present at the same time.

Explore

(15 Minutes)

Tell your students to use the *Explore* page of the Lesson Presentation to guide them through this challenge.

In addition, you can use these steps to direct them:

- **Step 1:** Ask your students to think about a simple dance move they can teach someone.
- **Step 2:** Have your students write down the steps needed to perform their dance moves. Explain that this will be their pseudocode.
- **Step 3:** Divide your students into pairs. Explain that partner 1 will share their pseudocode with partner 2. Partner 2 must follow partner 1's pseudocode exactly as they've heard it. Then, have them switch roles.
- You can also try role-playing this activity as a class. Ask all of your students to write pseudocode for a dance move, and then have them take turns sharing their pseudocode while the rest of the class tries to perform the move.

Hybrid Learning Tips for the *Explore* Phase

Make this a social time. Students enjoy working as a community.

Synchronous

- If your students are participating online, they can work together in one meeting room with one student sharing their pseudocode with the rest of the class. Alternatively, have them work in small groups in breakout rooms and then share their pseudocode with a partner or within their groups.

Asynchronous

- With family help, encourage your students to create a designated workspace for at-home activities. A well-lit, flat, clean surface that's free of clutter makes an optimal working surface. Please keep in mind that every student's home situation will be different.
-

Explain

(10 Minutes)

- Once your students have completed the coding challenge, have them share their experiences with a partner.
- The *Explain* page of the Lesson Presentation provides conversation prompts to help students give and receive feedback.
- Remind your students to use eye contact and a friendly tone of voice, and to focus on specific examples when giving feedback. Encourage them to reflect on how they can improve their work based on the feedback they receive.

Here are some suggested prompts for giving feedback:

- I'm really impressed with...
- I'd like to know more about...
- I had the same idea...
- I had a different idea...
- I suggest...

Here are some suggested prompts for receiving feedback:

- Which instructions were clear?
- What could I improve?
- What bugs can be fixed?

After your students have shared feedback, ask them to explain how this activity is like programming.

Ask questions like these:

- How is this activity like programming? (*Sample response: We're writing directions for someone to follow, just like when you write code for a computer to follow.*)
- Why write pseudocode? (*Sample response: to help you plan code before you write it on the computer so you can anticipate potential bugs*)
- What details are important to keep in mind when writing pseudocode? (*Sample responses: the importance of detail; the steps being in the right order*)

Hybrid Learning Tips for the *Explain* Phase

Synchronous

- If your students are participating online, review the feedback conversation prompts on the *Explain* page of the Lesson Presentation, and have them give and receive feedback in small groups in breakout rooms.
- Visit the breakout rooms to check for student understanding by having each group share a piece of specific feedback.
- Answer your students' questions and directly address misconceptions to develop a deeper understanding of the concept at hand.

Asynchronous

- Encourage your students to reflect on the feedback they've received by sharing back with the whole group or by posting a comment in the group chat.

Elaborate

(5 Minutes)

- Ask your students to brainstorm how they could improve their pseudocode or code different moves (e.g., their morning routine, a sports play, a recipe) to try this activity again.

Ask questions like these:

- How would you improve your pseudocode?
- How would you try this activity again?
- What are some other activities you could write pseudocode for?

If time allows, display the *Elaborate* page of the Lesson Presentation, which shows the LEGO Education SPIKE App programming blocks.

Use these talking points to continue the discussion:

- What are these programming blocks used for? (*Sample responses: to repeat the same action 10 times; wait one second before the action starts; if one action happens then another action happens*)
- Ask your students to share an example of a repeat, wait for, and if-then command in their dance moves.

in their dance moves.

Hybrid Learning Tips for the Elaborate Phase

Provide opportunities for your students to display their work and see what others have done.

Synchronous

- Host a scheduled online sharing session in which your students share how they might revise their inventions.

Asynchronous

- Set up a virtual gallery walk where your students can share their work. Encourage them to post pictures and videos of their work and check out their peers' work.
-

Evaluate

(5 Minutes)

- Invite each student to share an example of how their pseudocode showed step-by-step instructions (i.e., their algorithm), and to show an example of where they identified and fixed a problem in the pseudocode.

Ask your students to self-evaluate their performance. Use the suggestions below, or invent your own self-evaluation activity.

- Hold up one hand if you could use some help with writing pseudocode and debugging a program.
- Hold up two hands if you can explain how to write pseudocode and debug a program.
- Wave your hands if you can teach others how to write pseudocode and debug a program.

Hybrid Learning Tips for the Evaluate Phase

You have many options for assessment— a written assignment, a live or recorded oral presentation, or a drawing that demonstrates your students' understanding.

Synchronous

- Facilitate a whole-class virtual discussion during your scheduled lesson time.

Asynchronous

- Have your students reflect on their work by posting a written assignment, an oral presentation, or a drawing that demonstrates their understanding.

