

code jumper



Lesson 8
Decomposition

Code Jumper Curriculum: Lessons



Lesson 8 Decomposition

American Printing House for the Blind

Code Jumper Curriculum: Lessons

Copyright © 2020 American Printing House for the Blind

All rights reserved. No part of this publication may be reproduced, stored in retrieval system, or transmitted in any form or by any means, except as expressly permitted under copyright law, without written permission of the publisher.

Published by American Printing House for the Blind

1839 Frankfort Avenue, Louisville, KY 40206

www.aph.org | info@aph.org

OVERVIEW

LESSON OBJECTIVES

Students will:

- Identify a complex problem that will be easier to solve by breaking it into smaller problems
- Understand that one method for solving a complex problem is to break it down into smaller problems, making it easier to solve

EXPECTED OUTCOMES

Students will:

- All students: Be able to identify a complex problem that will need to be broken down into smaller problems to solve it
- Most students: Be able to break down a complex problem into smaller parts to make it easier to solve
- Some students: Be able to break down a complex problem that is made up of three or more parts

LESSON PLAN STRUCTURE

- Unplugged Activity
- Guided Code Jumper Activity
- Exploration
- Standards and Check for Understanding

RESOURCES

- Code Jumper Tutorial Videos
 - The Hub: <https://www.youtube.com/watch?v=KGb51PW9zJQ&lis=>
 - Play and Pause Pod: <https://www.youtube.com/watch?v=446jCw8qcDI&t=>

- Loop Pod: <https://www.youtube.com/watch?v=EFKbKLID3HI>
- Code Cards

KEY VOCABULARY

Decomposition: A key strategy within computer programming where you need to break down a complex problem into smaller parts in order to make it easier to manage, understand, and solve

SAMPLE PROGRAM SET UP

- Sample program set up and run by the teacher: Melody and Rhythm 1

- Sound Set: **MIDI Instruments : Piano** at Thread 1,
THREAD 1 Piano

LOOP 3 times

PLAY C5 for 1/2 a beat

PLAY D5 for 1/2 a beat

PLAY E5 for 1/2 a beat

END LOOP

END THREAD

- **Sample Sounds: Percussion** at Thread 2

THREAD 2 Percussion

LOOP 8 times

PLAY Hi-Hat 1 for 1.5 times speed

PLAY Hi-Hat 2 for 1.5 times speed

END LOOP

END THREAD

UNPLUGGED ACTIVITY: BREAKING BIG JOBS INTO LITTLE JOBS

OBJECTIVE

Introduce students to the key strategy for completing a more complex task by breaking it down into smaller tasks.

(Teacher background information) The concept of decomposition is key in solving larger problems. One of the key skills of software engineers is their ability to take large, complicated problems and break them down into smaller problems to solve. These smaller problems are, in turn, broken down even further until the problem reaches a point where it is more manageable. After the smaller problems are solved, they are then used to solve the bigger problems and the problem is built back up the chain, incorporating more and more of the solved smaller problems until the original problem has a solution. In bigger projects, the software engineer may not even know what the original problem is; they will only be concerned with solving the smaller problem that they have been given.

VOCABULARY

- Decomposition: A key strategy within computer programming where you need to break down a complex problem into smaller parts in order to make it easier to manage, understand, and solve

MATERIALS

- One cardboard box with rocks in the classroom
- One empty cardboard box in the hallway
- Rocks

- At least one rock (approximately the size of a baseball) per student
 - Timer
 - Computer Science Journal

INSTRUCTIONS

1. Prior to starting the Unplugged Activity, place one box containing the rocks somewhere in the classroom where all students can gather around it. Place an empty box in the hallway.
2. Have students gather in a circle around the box of rocks in the classroom, ensuring the box is at the center of the students.
3. Explain to students that their task is to figure out a strategy that will be the fastest and most efficient way to empty all the rocks from the box in the classroom and fill up the box in the hallway.
 - a. Conditions students must follow:
 - i. Each student may only take one rock
 - ii. Each student can hold only one rock at a time
 - iii. Students can only walk (no running) in the classroom
 - iv. Students may hand a rock to another student
 - v. If a rock falls on the floor, the rock must go back to the classroom box and the student must start their turn again
4. Instruct students to come up with an initial strategy to try, and time them carrying out their plan to get the rocks from one box to the other. (Example: Students may decide to each pick up a rock one after the other from the classroom box and quickly walk to the hallway and put it in the other box.)

- a. Strategies for students with a visual impairment:
 - i. Run a string from the box of rocks to the box outside the classroom for the students to follow.
 - ii. Have some students be "spotters" along the way.
 - iii. Students can give each other very clear directions to meet the next person. (Example: "Walk three steps forward and one step to the right.")
5. After coming up with an initial amount of time for the activity, debrief with students what worked well and what could be modified to make it more efficient.
6. After the discussion, instruct students to figure out another strategy they might like to try. They might also decide to modify the strategy they initially used. (Example: Students line up from one box to another, and the first person picks up the rock and passes it to the person standing next to them, creating more of an assembly-line strategy.)
7. When the second attempt at a strategy is completed, compare the two times to see which strategy was quicker and more efficient.
8. Discuss with students why one strategy worked better than the other.
9. If needed, repeat the process of coming up with another strategy and then debrief.
10. Once the class has tried three strategies, review which one allowed them to move the rocks from one box to the other the most efficiently.
11. Explain to students that the activity allowed them to see how the large problem of transferring heavy rocks from one box to another would be made easier by developing a key strategy focused on breaking a big task into smaller ones in order for it to be completed.

CLOSURE

Review with students that in computer programming, **decomposition** is a key strategy where you need to break down a complex problem into smaller parts in order to make it easier to manage, understand, and solve.

GUIDED ACTIVITY: DECOMPOSITION

OBJECTIVE

Students will break down programs into smaller, manageable pieces.

VOCABULARY

- **Decomposition:** A key strategy within computer programming where you need to break down a complex problem into smaller parts in order to make it easier to manage, understand, and solve

MATERIALS

- Decomposition Guided Activity Melody and Rhythm 1 Code Card:
- Decomposition Guided Activity Melody and Rhythm 2
- Computer Science Journal

INSTRUCTIONS

1. Create the Decomposition Guided Activity Melody and Rhythm 1 program:
Tip! Instead of using notes, use a simpler, easy to remember sequence of sounds, such as those in the Nature, Numbers, or Animals Sound Sets, or a custom Sound Set that is familiar to students.

2. Instruct students to listen to the Melody and Rhythm 1 program. It is important that they are not tracing the code or seeing the physical program. Discuss the idea that this program sounds complex and that there is more than one thing happening at a time. Ask students: How do we know what is going on? How could we build this program in Code Jumper after listening to it?
3. Ask the students to brainstorm and think of things that they do every day that require them to break things down into smaller steps. Ask students: When you want to create, build, or draw to accomplish a task, do you know how to do it, or do you need to figure out what steps to take? How about when something is not working quite right? What process do you go through to figure out what to do?
4. Have students write in their Computer Science Journals something that requires them to break a problem down into smaller steps. (Examples: Making a sandwich, tying their shoes, getting ready for school, etc.)
5. Play the program again and ask students questions about what they are hearing. How many threads? (Two) How many sounds? (Five different sounds) What are the different sounds? How are we going to figure this out?
6. This process is known as **Decomposition**.
7. Listen to the program again. Ask students: How could we decompose the sample program Melody and Rhythm 1? (Expected response: Work on one thread at a time.)
8. Ask students to get out their Computer Science Journals and write down what they are hearing.
9. Ask students: How many threads do you think there are in this program? (Expected response: Two.)
10. Ask students to decide which thread they are going to work on first—piano or percussion.

11. Run the program again. Ask students to concentrate only on the thread that they have just chosen, recording the sounds that they hear in the Computer Science Journal.
12. Run the program again and ask the students to focus on the other thread.
13. Ask students to recreate the thread in Code Jumper. Run the program again and ask students to debug the thread if there are errors. Ask students to write down the code in their Computer Science Journals.
14. Once students are happy with their first thread, instruct them to move on to the second thread and recreate it. Again, instruct students to write down their code in their Computer Science Journals.
15. Ask students to compare their complete program with their original program and make any necessary changes. For a final check, remove the other thread from the Hub. Do not take apart all the Pods, but just remove the connection at the Hub.
16. Play the sample program Decomposition Guided Activity Melody and Rhythm 2. (Code Card located at the end of the lesson)
17. Ask students: How are you going to solve this problem? (Expected response: By decomposing it.)
18. Ask students to work in their groups to recreate the program. Ask students to raise their hand if they would like to have the original program played again.
 - How many threads are there? (Expected response: Two.)
 - Decide which thread they want to work on first, and then have them recreate it.
 - Move on to the second thread and recreate it.
 - Listen to their programs, compare them to the original one, and debug, if necessary.

CLOSURE

Ask students to explain how they can solve a complex problem and to give an example. Instruct students to record their response in their Computer Science Journals.

EXPLORATION

OBJECTIVE

Students will explore decomposing (taking apart) programs that have been created by their peers and record their observations in their Computer Science Journals.

VOCABULARY

Decomposition: A key strategy within computer programming where you need to break down a complex problem into smaller parts in order to make it easier to manage, understand, and solve

MATERIALS

- Code Jumper Kit
- Computer Science Journal

INSTRUCTION

1. Students will need to write a program in their Computer Science Journals using:
 - a. A total of six Play pods
 - b. A loop
 - c. Two threads
2. Ask the groups of students to create the program in their Computer Science Journal and in Code Jumper.
3. Once the program in Code Jumper has been debugged and matches the written code, students will then play the program for another group to decompose.

4. Strategies for this can be:
 - a. Place a piece of paper or cloth over the Pods with only the Hub visible so the program can be listened to multiple times.
 - b. Turn the computer or tablet around so students cannot view the code on the screen.
5. Instruct the groups of students to write the code in their Computer Science Journals.
6. Once the groups have finished, have the groups share with each other to determine if they have correctly decomposed the program.

CLOSURE

Ask students to reflect on their experience in their Computer Science Journals.

STANDARDS AND CHECK FOR UNDERSTANDING

CSTA K-12 COMPUTER SCIENCE STANDARDS *

- 1B-AP-11: Decompose (break down) problems into smaller, manageable sub problems to facilitate the program development process.

NATIONAL CURRICULUM OF ENGLAND*:

Key Stage 1:

- Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- Create and debug simple programs

- Use logical reasoning to predict the behavior of simple programs
- Use technology purposefully to create, organize, store, manipulate and retrieve digital content
- Recognize common uses of information technology beyond school
- Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

Key Stage 2:

- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- Use technology safely, respectfully and responsibly; recognize acceptable/unacceptable behavior; identify a range of ways to report concerns about content and contact.

CLOSING ACTIVITIES AND CHECK FOR UNDERSTANDING

Ask students to think about other areas in their lives where they decompose tasks or activities.

Check for Understanding	Completed
Student can examine a program that is complete and break it down into the individual components.	Yes / No

*Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from <http://www.csteachers.org/standards>

*Education, Department for. "National Curriculum in England: Computing Programmes of Study." GOV.UK, 11 Sept. 2013, www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

CODE CARDS

DECOMPOSITION GUIDED ACTIVITY MELODY AND RHYTHM 1 CODE CARD:

THREAD 1 Piano

LOOP 3 times

PLAY C5 for ½ a beat

PLAY D5 for ½ a beat

PLAY E5 for ½ a beat

END LOOP

END THREAD

THREAD 2 Percussion

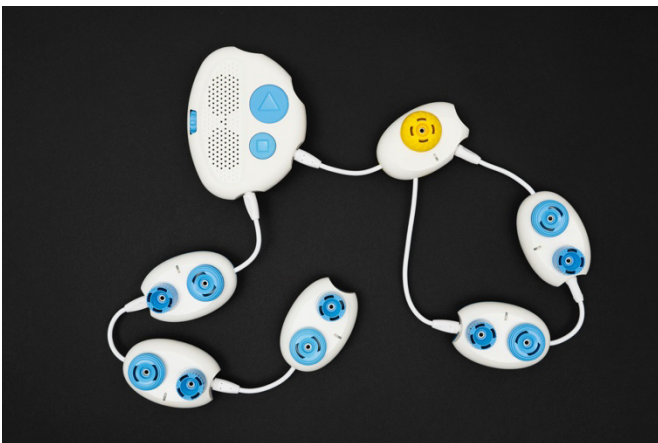
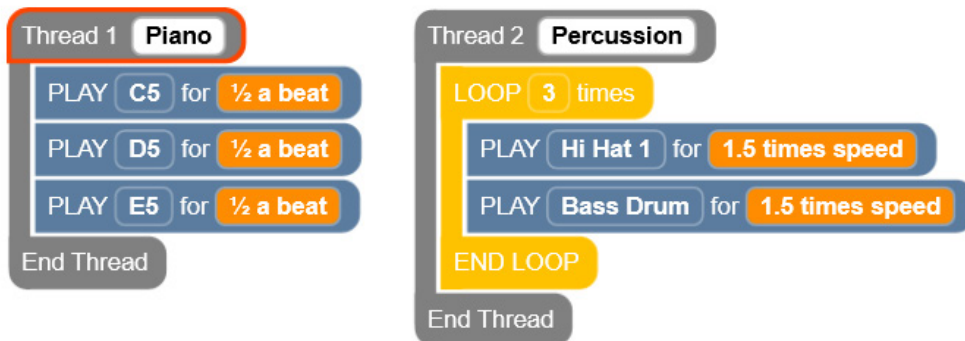
LOOP 3 times

PLAY Hi-Hat 1 for 1.5 times speed

PLAY Hi-Hat 2 for 1.5 times speed

END LOOP

END THREAD



[Figure Caption:] At the top is a screenshot of the Code Jumper app with two threads. Under Thread 1 Piano, the code reads, PLAY C5 for 1/2 a beat, PLAY D5 for 1/2 a beat, PLAY E5 for 1/2 a beat; the commands are followed by End Thread. Under Thread 2 Percussion, the code reads, LOOP 3 times, PLAY Hi Hat 1 for 1.5 times speed, PLAY Bass Drum for 1.5 times speed, END LOOP; the commands are followed by End Thread. Below this screenshot, three Play pods are connected to the Hub at Port 1. One end of a Loop pod is connected directly to the Hub at Port 2 and is also connected to two Play pods.

DECOMPOSITION GUIDED ACTIVITY MELODY AND RHYTHM 2

THREAD 1 Piano

LOOP 4 times

PLAY C5 for a $\frac{1}{4}$ beat

PLAY D5 for a $\frac{1}{4}$ beat

PLAY C5 for a $\frac{1}{4}$ beat

END LOOP

END THREAD

THREAD 2 Percussion

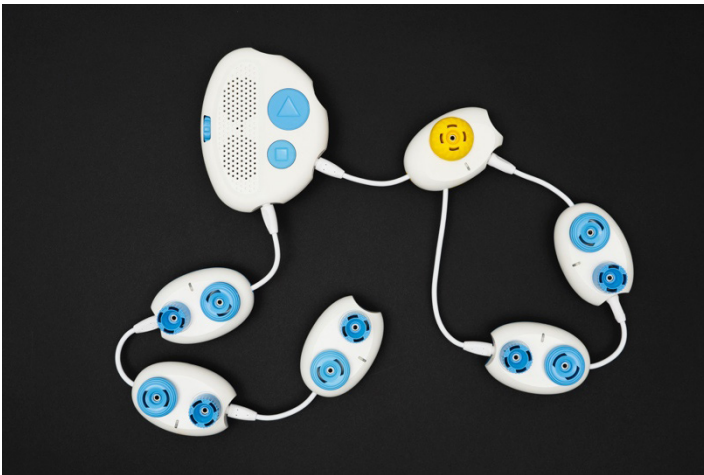
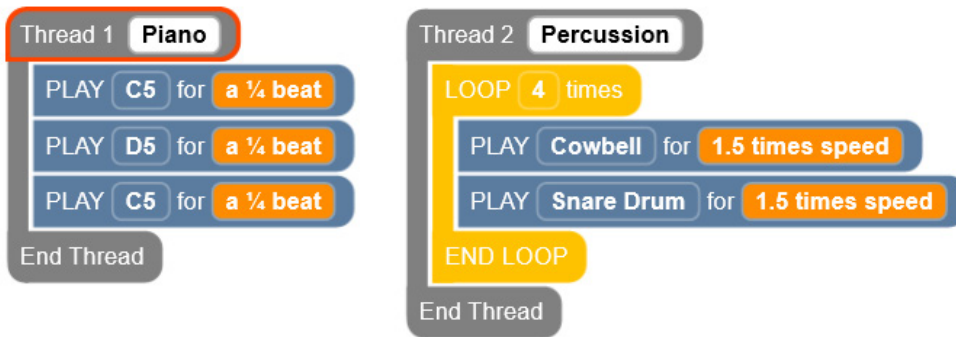
LOOP 5 times

PLAY Cowbell for 2 times speed

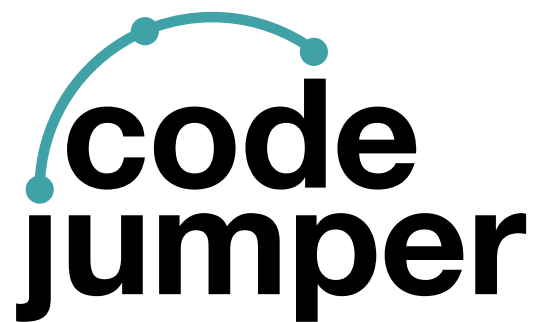
PLAY Snare Drum for 1.5 times speed

END LOOP

END THREAD



[Figure Caption:] At the top is a screenshot of the Code Jumper app with two threads. Under Thread 1 Piano, the code reads, PLAY C5 for a 1/4 beat, PLAY D5 for a 1/4 beat, PLAY C5 for a 1/4 beat; the commands are followed by End Thread. Under Thread 2 Percussion, the code reads, LOOP 4 times, PLAY Cowbell for 1.5 times speed, PLAY Snare Drum for 1.5 times speed, END LOOP; the commands are followed by End Thread. Below this screenshot is a photo of a Code Jumper program. Three Play pods are connected to the Hub at Port 1. One end of a Loop pod is connected directly to the Hub at Port 2 and is also connected to two Play pods.



For more resources, visit codejumper.com

Copyright © 2020



**AMERICAN
PRINTING
HOUSE**

**1839 Frankfort Avenue
Louisville, Kentucky 40206
502-895-2405 • 800-223-1839
www.aph.org • info@aph.org**