

Introduction to Debugging, Decomposition, and Algorithms

Drawing from directions (15–20 minutes)

This activity will introduce students to three big ideas in computational thinking (CT): debugging, decomposition, and algorithms.

Materials: Two simple line drawings or designs. Examples are given below the activity directions. You can make adjustments to suit the needs or interests of your class.

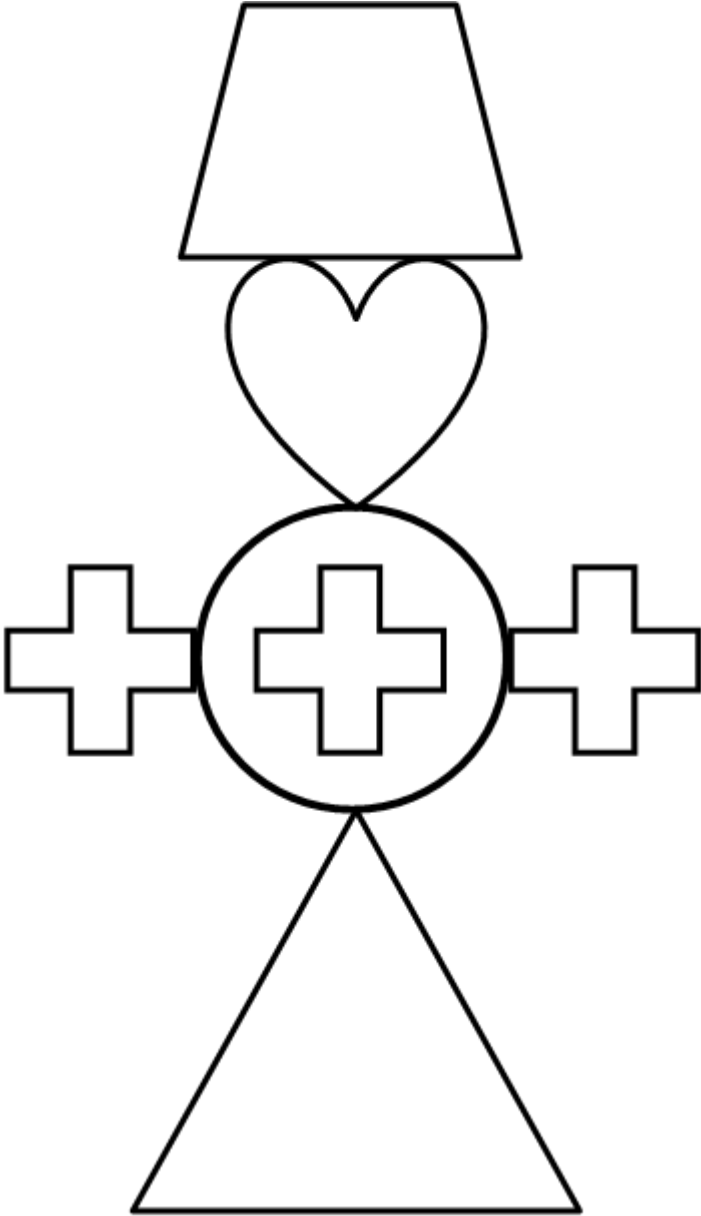
Directions

1. Divide your class into two groups. Show each group one of your two chosen designs. Students should not see the design used by the other group.
2. Tell students that their task is to write directions for how to draw their design. Someone who has not seen the design should be able to follow their directions to reproduce the design.
3. When students are finished writing their directions, have each student partner with someone from the other group. Students exchange directions. Without looking at the original designs, students should follow their partners' directions closely to create a drawing.
4. Students return the directions, along with the drawings they created, to their partners. Together with their partners, students compare the drawings with the original design. They talk together about ways they could change the directions so that the drawing produced would end up closer to the original design.
5. After students discuss improvements to their directions, bring the class back together for a discussion. Ask questions such as the following:
 - a. What were some differences between the original design and your partner's drawing?
 - b. What improvements did you make in your directions?
6. Explain that when students compared the drawings and made improvements to their directions, they were doing a kind of CT strategy called **debugging**. Debugging is a kind of problem solving where you check whether the outcome of your work matches what you intended. If the outcome or solution doesn't make sense or look like you intended, then there is a **bug**, or error, in the solution. When you try to figure out the bug and fix it, you are debugging. Tell students that they will be doing lots of debugging in math class this year, and that looking for and fixing errors is a normal and important part of being a good problem solver.

7. Next, ask students to talk to their partners about how they went about creating their directions. Questions that can guide their discussion include:
 - a. What part of the design did you decide to write directions for first?
 - b. Did you think about the whole design at once, or think about it in pieces?
8. Have a few volunteers describe how they went about writing their original directions. Expect that some students will discuss handling one part of the design at a time. Explain that these students were doing another kind of CT strategy called **decomposition**. Decomposition is a process of breaking a complex problem or situation into parts to make it easier to solve. Ask students to discuss the following questions:
 - a. How might breaking a design into parts make writing directions easier?
 - b. What are the different ways you could decompose the design?Tell students that they will use **decomposition** as a strategy for solving challenging math problems this year. Explain that skilled problem solvers use this strategy to get started on and solve challenging problems.
9. Last, ask students to imagine that they had more time to debug and test their directions, and that they ended up with a very clear set that helped anyone produce the design exactly. Ask students to share their experiences by following detailed directions, such as when they follow recipes or directions for assembling a model or toy. Ask:
 - a. Why is it useful to have a clear set of steps for completing a task?
10. Explain that a clear, step-by-step process for solving a problem is called an **algorithm**. Developing and following algorithms is another important kind of CT strategy that students will use in math class this year. Tell students that they will be doing a lot of thinking about how to create algorithms and why algorithms work. This will help them become powerful problem solvers.

(Sample designs are on the next two pages.)

Sample design 1



Sample design 2

