Recognizing & Defining Computational Problems

This practice encourages students to solve a problem by first defining the problem, breaking it down into smaller parts, and then evaluating each part to determine if a computational solution makes sense to use. The goal is help students understand that dealing with many different stages all at once is more difficult than breaking a problem down into several smaller problems and solving each one.

Students should...

**Identify complex, interdisciplinary, real-world problems**

Students should be able to identify real-world problems that span multiple disciplines and can be solved computationally, such as designing new bike helmets to increase safety or creating a tool to harvest food more efficiently.

**Decompose complex real-world problems**

Students should be able to break problems down into their component parts. They should decompose complex problems into manageable sub-problems that could potentially be solved with clear programs or procedures.

**Evaluate whether computational solutions are appropriate and feasible**

As students progress, they should evaluate the feasibility of using computational tools to solve given problems or sub-problems, such as through a cost-benefit analysis.

**Sample Student Task:**

Task your students with developing a mobile phone application to solve a problem in their world. This is a complex task - there are lots of things to consider. How would you decompose the task of creating an app? Rather than have students jump into creating an app, have them decompose the problem into smaller sub-problems and determine if using computational tools are appropriate for each of those sub-problems.

**Resources:**

- An Introduction to Decomposition
- Decomposition Sample Lesson Plan
- Navigating the Practices Framework

Source: K-12 Computer Science Framework