Developing and Using Abstractions

Abstractions are formed by identifying patterns and extracting common features from specific examples to create generalizations. Using generalized solutions and parts of solutions designed for broad reuse simplifies the development process by managing complexity.

Students should...

Extract common features from a set of interrelated processes or ideas

Young learners should be able to identify and describe repeated sequences in data or code through analogy to visual patterns or physical sequences of objects. As they progress, students should identify patterns as opportunities for abstraction, such as recognizing repeated patterns of code.

Evaluate existing technological functionalities

Students should be able to use well-defined abstractions that hide complexity. Just as a car hides operating details, such as the mechanics of the engine, a computer program’s “move” command relies on hidden details that cause an object to change location on the screen.

Create simple modules and points of interaction

As they progress, students should take advantage of opportunities to develop generalizable modules. For example, students could write more efficient programs by designing procedures that are used multiple times in the program.

Sample Student Task:

Students can watch animations on Youtube that represent key grade-level concepts, such as the water cycle or a rocket launch. Students evaluate the effectiveness of the animation in explaining the process. Students create their own stop-motion animation sequences to better represent the scientific phenomenon they studied.

Resources:

- Abstraction with Mad Glibs Lesson Plan
- Explaining User Experience (UX) to Kindergartners
- K-12 Computational Thinking Practices in Action

Source: K-12 Computer Science Framework