Practice	Definition	Rationale	Questions
Planning	Discussing and representing problems visually before attempting to solve them.	Students are often inclined to start coding without a clear plan in mind. It's important to pause and plan before jumping in.	• What is the problem you're trying to solve? • What have you done so far (if anything)? • Would it help to read the instructions or to draw a picture?
Decomposition	Breaking down complex tasks into smaller, more manageable pieces.	Large tasks often seem insurmountable at first. The key is to break them down and build and test solutions incrementally.	• Can the overall task be broken into smaller pieces? • What's the first piece you could build and test? • What will come second, third, fourth, etc.?
Prediction	Formulating and articulating expectations for how a program will work.	Every time students run their code, they should have a hypothesis about what will happen.	• What do you expect your code to do? • How will you know if your code is working (or not)? • If it doesn't work, what will you try next?
Observation	Carefully watching to see how a program actually performs.	To see whether code is working, students need to observe carefully, keeping test conditions as consistent as possible.	• What do you see happening when the code runs? • How does what you see differ from what you want? • Are your testing conditions identical every time?
Debugging	Identifying and resolving problems (i.e. "bugs") in a computer program.	A program rarely works perfectly the first time. Coding is an iterative process of testing, debugging, and refinement.	• Where in your program does the problem occur? • Can different sections be tested in isolation? • What could you change to create more evidence?
Abstraction	Noticing similarities among pieces of a program or computing challenge.	Often, a first attempt to complete a computing task will reveal patterns that can be used to simplify the code.	• Does this program contain repeating patterns? • Do any sprites or objects behave in similar ways? • Could loops or functions be used to simplify the code?
Collaboration	Working and communicating with others to solve problems.	Coding is not a solitary pursuit. It's important for students to be able to work with others, and discuss problems and solutions.	• How can you divide responsibilities efficiently? • How would you explain your code to a classmate? • What is a challenge you encountered and overcame in this project?