A free, open-access 4-part teacher PD design by Grover & Twarek (2023) to help—

2. Appreciate the characteristics of good formative assessment.
3. Expand their repertoire of types formative assessment used and understand the use and value of different types.
4. Understand the many kinds of quick, auto-grading assessment types for use while teaching programming.
5. Evaluate assess items using a framework or set of design features.
6. Identify common misconceptions in introductory programming and select/create items to diagnose learning.
7. Plan how to respond to assessment results.

A Guiding Framework for Helping Teachers Evaluate & Create Formative Assessments for K-12 CS Classrooms

Pellegrino, DiBello, and Goldman (2016) proposed a framework for conceptualizing and evaluating the validity of classroom assessments to address the need for examining assessments intended to function close to instruction.

Cognitive: how well does the assessment align to a domain knowledge/skill in ways that are not confounded with other aspects of cognition such as language or working memory load?

Inferential: how well is an assessment aligned with instructional content, including students’ opportunities to learn?, does it provide instructionally useful information?

Extending the Framework to Address Fairness & Equity and Guiding Questions for Evaluating Formative Assessments Quality

1. Does the assessment require other knowledge not directly related to the CS topic (for example, some mathematics or domain knowledge of another subject that the students have not learned?)
2. Is the assessment clear in language and design (i.e., Are the language and question unambiguous)?
3. What constructs/concepts/topics is this assessment addressing? Are they obvious to you from examining the assessment?
4. Are these constructs/concepts/topics part of a (A) Learning goals or (B) CSTA standards for this grade level?
5. Did students have an opportunity to learn this in the classroom (i.e., did I cover it adequately in my teaching)?
6. If a student gets the answer wrong, are you able to tell exactly where they have a gap in their understanding?
7. Is there enough clarity from the response on follow-up action if student(s) get(s) it wrong?
8. Is the assessment designed at a level that is appropriate for all your students or does it privilege or disadvantage some? (for example, are there cultural references that some students may not know about that are essential to tackling the assessment?)
9. If not, are you intending to gather information about who can tackle it and who cannot. If yes, then it’s ok to use this assessment, else, ensure that differently abled students or students from diverse backgrounds are provided opportunities to demonstrate their understanding in alternate ways.

Putting the Formative Assessment Evaluation Framework to Work

Cognitive: This is a good assessment as it clearly targets when/how value variables change/are updated.

Instructional: Variables are a key concept.

Inferential: Excellent! Diagnostic question and points to what must be done if students do not get this right. (For example, step through code and inspect variables, demonstrate the use of a ‘if’ and ‘while’ block).

Cognitive: This assessment targets knowledge of the coordinate system for the Code.org programming environment "stage" as well as basic JavaScript animation functions; neither is a generic domain skill for programming (same for Scratch stage-related Qs).

Instructional: Presumably this content is covered when introducing learners to the programming environment.

Inferential: Weak! Since this assessment involves an understanding of basic JavaScript functions to create and animate a sprite, as well as the x/y coordinate system of the stage, an incorrect answer could point to a lack of understanding of either of these.

Cognitive: This assessment targets while: loops work, expressions control loops, and update of variables.

Instructional: Loops are a key concept in programming.

Inferential: Even though 2 concepts are targeted, the responses provide a clear indication of where a student is lacking understanding. It targets a common misconception students have about ‘while’ loops.

Cognitive: This assessment targets how: while loops work, expressions control loops, and update of variables.

Instructional: This is a key part of teaching introductory programming and should score high on that.

Inferential: Even though 2 concepts are targeted, the responses provide a clear indication of where a student is lacking understanding. It targets a common misconception students have about ‘while’ loops.

References
