Hint Rating Rubric

The rubric on the next page was used to rate hints from a Fall 2016 deploy of the iSnap programming environment. The full dataset is available on the PSLC Datashop: https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1920

Additionally, the hints ratings themselves can be found under Files->First and Second Hint Ratings. These ratings can be matched to hints in the dataset using the assignment ID, the submission/project ID and the timestamp.

Ratings were made while looking at the student's code at the time of the help request, along with the content of the hint dialog. Raters had access to the student's code history as well. A sample interface, using public data, can be found here: http://arena.csc.ncsu.edu/public/snap/hints/view/

Relevance: Think of all the goals that the student may have had when they asked for the hint, based on what they have accomplished, the errors that they have and what they have left to do. How likely is it that the hint they received addresses one of these goal?

1. The hint is not at all relevant to the student's likely goals.
2. The hint is somewhat relevant to a likely goal, or addresses a goal that is possible but not likely for the student to be pursuing.
3. The hint is directly relevant to a likely student goal.

Correctness: Will following this hint bring the student closer to correctly completing the assignment, by progressing the code or removing erroneous code? Note that if the student has a non-traditional solution, the hint should be judged with respect to the student's code, not an "ideal" solution.

1. The hint suggestions is either not useful or removes useful code.
2. The hint suggestion is technically or partly correct, but may not be fully necessary. For example, the hint may suggest a better solution for code that is already correct. Alternately, the hint may contain some edits which are useful and others that are not.
3. The hint moves the student towards a correct solution, by adding useful code or removing erroneous code.

Interpretability: How easy is it for a novice to understand the value of the suggestion?

1. The hint is either incorrect or the value of the suggestion would not be clear without further explanation. Following this hint would not likely get the student "unstuck" without further hints.
2. The value of the hint may or may not be apparent to the student. For example, the hint may be too vague, or might come too early for the student to understand it.
3. The hint should be easy to interpret and understand, assuming the student understands the objectives of the assignment.

Important note: These ratings should be largely independent. Hints can be:

- Correct and interpretable but not relevant: such as saying to add a variable at the start of a program, when the student has no code to use the variable.
- Relevant but not correct: such as telling the student to add a needed block in the wrong place.
- Relevant and correct but not interpretable: such as telling the student to delete a large chunk of bad code that's causing a bug, without explaining what code to save or move elsewhere.

However, a hint that is incorrect may be uninterpretable as a result, since there is no value in it to understand.