

An Empirical Comparison of Rubric-Based and Conventional Assessment in Programming Education

Rafia Sultana¹, Maysha Fahmida Bristy²
^{1,2}(City University, Bangladesh)

Abstract: Assessment in programming courses is not about checking if a program gives the right answer. It also looks at how students think through problems how they write and organize their code how they explain their work and how well they use computer resources. This study compares two ways of grading: one using a rubric and the other using the instructor's own judgment. The task was part of a Data Structure Laboratory. It involved finding the smallest number in an array. The rubric-based method checks students' work against categories: whether the logic is correct how clean and well-structured the code is, if the output is accurate how well the code is documented and how efficiently it runs. The study found that with the rubric the average score was 7.6 (with a deviation of 0.8). With the method the average was 6.8 (with a standard deviation of 1.2). So students got scores on average with the rubric and the scores were more similar across students. Less spread out.

Keywords: programming education; analytical rubric; assessment; grading consistency; data structures

Date of Submission: 15-10-2026

Date of Acceptance: 28-10-2026

I. Introduction

Programming education asks students to show they have related skills. A student might get the result but use a slow way to do it write messy code or not explain their work well. So the way programming work is graded needs to look at more than the final answer. How programming tasks are judged can also affect what students understand about what's expected and how they can get better.

Traditional grading mostly depends on what a teacher thinks about a students work. This kind of judgment can be helpful especially when looking at creative projects.. If the rules aren't clear students might not know how their grades are decided. Also when different teachers see correctness, code quality and explanations it can make grading feel less fair.

A rubric-based approach gives an organized way to grade. It sets out the things to look for. When needed shows what different levels of performance look like before anything is graded. In programming classes a rubric can break down parts like how the logic works how clean the code is, how accurate the output is, how well it is explained and how efficient it is. This structure can help teachers explain what is expected and give specific feedback. Earlier studies on rubrics in college classes have talked about how they can help with being clear giving feedback and making grading better.

Even though there are things about using rubrics how well they work depends on how they are made how clear the descriptions are and how consistently they are used. Comparing rubric-based grading with methods can then give useful information about how organized grading works in a certain programming class.

This research looks at the scores from rubric-based and traditional grading for a programming task, in a Data Structure Laboratory class. The study is based on two questions: (1) How do the scores differ between the two ways of grading? (2) What might clear grading rules mean for how clear and fair programming grading is? The results only apply to the task the people involved and the data that was actually gathered.

II. Literature Review

The poems that emerged in the post-pandemic era largely embrace themes of enthusiasm, optimism, and hope. This paper examines only four poems. These poems were chosen based on their shared themes. These poems were collected from various social media platforms: "Goodbye Pandemic" by Euis Cucu Sukmanah and "Face-to-Face Learning (PTM)" by Didi Prasetyobudi; "Rise Together with Corona" by Parni Hadi; and "Generasi Bangsa Rises to Work" by Sheli Rahmadina. These poems were analyzed using isotopy theory.

Isotopy theory was developed by Greimas. This theory is typically used in the analysis of utterances. The act of presenting events sequentially through sentences is called utterance. Genette (Zaimar, 1991) proposed the concept of mode and speech. Within the category of speech, narrative is discussed, encompassing the narrator, while the narrative's traces are described through pronouns, determiners, variations in lexical fields, and various syntactic functions. Isotopy lies in the variation of lexical fields. Lexical fields are cultural clues that can explain the narrator's background and can reveal hidden ideas that are at the heart of a work. Greimas, within his structural semiotic framework, introduced the concept of isotopy as a semantic category of repetition that examines the coherence of a text. This means that a poem is not only coherent because it is grammatically correct, but also because it contains a consistent "path of meaning." This path is deliberately formed by small components of meaning called seme, which are the repetition of 2.1 Rubrics in assessment

A rubric is a tool used in assessment to outline the criteria for evaluating work. It shows what is expected and includes levels of performance for each criterion. There are two types: analytical rubrics and holistic rubrics. Analytical rubrics look at each aspect of performance separately while holistic rubrics give an overall score. Analytical rubrics are especially helpful for programming assignments because programming involves different qualities. These qualities include correctness, logic, readability, documentation and efficiency. These elements are. Should be judged on their own.

Stevens and Levi say that rubrics help teachers share expectations with students and make feedback clearer. Jonsson and Svingby look at how reliable and valid rubrics are and what impact they have on learning [1 2]. Reddy and Andrade also examine how rubrics are used in education. They talk about how rubrics can guide assessment and support student learning [3]. The research shows that rubrics can make the evaluation process more transparent.. Their effectiveness depends on how well they are designed and how they are used.

2.2 Assessment in programming education

Assessing programming assignments is not easy. There can be correct solutions and these solutions may differ in structure, efficiency or clarity. A grade based on whether the program runs may not reflect the quality of the logic or the overall implementation. At the time if the criteria are too vague it becomes hard to score fairly. A structured rubric can help by clearly identifying the key parts of the task that should be judged.

Studies in programming education have also looked at debugging behavior and automated assessment tools [4, 5]. These methods focus on parts of the evaluation process.. They all show that it is important to know exactly what is being assessed. A rubric does not automatically make scoring accurate or consistent. The descriptions in the rubric must be clear match the learning goals and be applied in the way every time.

2.3 Research focus

This study looks at a comparison between traditional judgment and rubric-based scoring for a basic task involving arrays. The findings should be seen as specific to the context. Not, as evidence that rubrics are always better. To build conclusions future research needs to use clearly described samples well-tested scoring criteria and statistical methods that can be repeated.

one lexeme after another. Saaffan & Mahmoud (2024) include repetition of other units such as phonetics (rhyme) and prosody (rhythm). For Saaffan & Mahmoud, isotopy is not simply the repetition of the same word, but rather a "direction of interpretation" that allows readers to organize semantic and lexical networks into a coherent network, while still considering the subject's involvement in it. Rejo (2014) also states that isotopy refers to the phenomenon where different expressions in a text share the same seme, thus creating a homogeneous meaning.

The theories of fields of meaning and lexical fields were pioneered by Trier. He stated that the vocabulary of a language is synchronically the entire system of lexemes interrelated in a single meaning (Lyons, 1977). Porzig also researched fields of meaning. Lyons then summarized the two theories by stating that lexemes that are paradigmatically and syntagmatically related can be said to be elements of the same field. For example, the lexemes "hit," "punch," and "hand" can be grouped into one field of meaning because all three are associated with the lexeme "hand."

Genette's procedure in lexical field analysis is to examine isotopy. Isotopy is a type of repetition. The word isotopy comes from the Greek isos, meaning "same," and topos, meaning "place" (Zaimar, 1991). Isotopy is an open area or field of meaning consisting of all elements found throughout a discourse/text. There is no specific criterion for determining isotopy in a work. Usually, just by grouping prominent words that do not need to have a generic relationship, but can be linked to human personal experience.

III. Methods

3.1 Study. Context

The study was conducted in a Data Structure Laboratory course at City University. Participants completed a programming exercise that required them to identify the element in a specified array. The original manuscript contains sample-size statements: the abstract refers to ten students whereas the methodology and conclusion refer to twenty students. The correct number must be confirmed from the attendance records. Dataset before this manuscript is submitted.

3.2 Programming task

The task required each participant to write a program that determines the smallest element in an array. This task was chosen to provide a programming problem for evaluating student solutions. The final manuscript should state the programming language, task prompt, execution environment, time limit (if any) and whether all students received instructions.

3.3 Assessment approaches

Under assessment the instructor assigned a score based on an overall judgment of the submitted solution. The source manuscript indicates that correctness, code structure and demonstration of the output informed the mark. It does not provide a complete scoring guide for this method.

Under rubric-based assessment the solution was evaluated against five dimensions: correctness, code quality, output accuracy, documentation and efficiency. The original manuscript indicates that marks were allocated to the dimensions and summed to produce a total score out of ten. Before submission the authors should insert the rubric, including the maximum score for each criterion and the descriptors used to distinguish performance levels.

3.4 Data collection and analysis

The original manuscript reports statistics for both assessment approaches and states that a t-test was used to examine the difference in scores. The analysis should be reproduced from the student-level data. If the same students received scores under both approaches a paired-samples t-test may be appropriate if its assumptions are sufficiently met; if the groups or observations were independent a different test may be required. The final report should identify the test, sample size, test statistic, degrees of freedom, exact p-value, confidence interval and effect size where possible. No unreported statistic should be. Fabricated.

The study should also clarify whether assessment order was controlled, whether the same submitted code was scored by both methods and whether, than one assessor participated. These details affect how the score comparison can be interpreted.

3.5 Rubrics Table

Table I : Rubrics Based Assessment Table

Criteria	Excellent (4)	Good (3)	Satisfactory (2)	Poor (1)	Marks
Problem Understanding	Clearly understands the task	Mostly understands	Partially understands	Cannot explain	2
Algorithm / Logic	Correctly identifies and applies minimum-finding logic	Minor logical issue	Partially correct	Incorrect logic	3
Code Implementation	Correct, complete and well-structured code	Minor coding errors	Several errors	Incomplete/incorrect	3
Output	Correct minimum value for all test cases	Mostly correct	Correct for some cases	Incorrect/no output	1
Explanation / Viva	Clearly explains the code and logic	Explains most parts	Limited explanation	Cannot explain	1

IV. Results

The source manuscript reports the following summary statistics. These values are provisional. Should be checked against the original data before publication.

Table II: Result

Assessment approach	Mean score	Standard deviation
Conventional assessment	6.8	1.2
Rubric-based assessment	7.6	0.8

The mean reported under rubric-based assessment is 0.8 points higher than the mean reported under conventional assessment. The standard deviation reported for rubric-based scores is 0.8 than the 1.2 reported for conventional assessment, which suggests less spread in rubric-based scores in this sample. These descriptive differences alone do not prove that the assessment method caused improvement or that the difference is statistically significant.

4.2 Inferential analysis

The source manuscript states that a t-test found a significant difference at $p < 0.05$. However the manuscript does not provide the p-value, test statistic, degrees of freedom, confidence interval or the underlying scores. In addition the sample size appears inconsistent. Must be resolved. Therefore the inferential result cannot be verified independently from the supplied document. Insert the statistical output here after rerunning the appropriate test, on the original data.

V. Discussion

The reported results show that rubric-based assessment had an average score and a smaller standard deviation compared to conventional assessment. One reason might be that clear criteria make it easier for graders to see what is being scored and apply the standards to each submission. For students having points like logic, code quality, output accuracy, documentation and efficiency makes it clear that programming isn't just about getting the right output. It's also about how well the code is written and explained.

We should be careful when interpreting these results. A higher average score doesn't always mean students learned more.. A lower standard deviation doesn't automatically mean grading is more reliable. The pattern we see might also be because of how each method was used, how the rubric was designed or the nature of this one task. The study would be stronger if than one person graded the same submissions and we could measure how much they agreed with each other.

There are also some limits to this study. The sample is small. Hasn't been confirmed yet. It only used one programming task. The manuscript doesn't include full student-level data or inferential statistics. We also don't know if both assessment methods were applied to the work under the same conditions. Because of these issues we can't be sure the results apply to courses, different programming tasks, other schools or different student groups.

Future studies could use an more varied group of students include several programming tasks with different levels of difficulty have multiple graders and use a rubric that's planned ahead of time. Researchers could also ask students for feedback on how clear and fair the assessment felt. That would help tell the difference between how students performed and how reliably the grading was done, as well as how useful the feedback was.

VI. Conclusion

This paper compares rubric-based and conventional assessment in a Data Structure Laboratory course. The report shows a score of 7.6 (SD = 0.8) for rubric-based assessment and 6.8 (SD = 1.2) for conventional assessment. These numbers suggest average scores and less spread in scores with rubric-based assessment in this sample.. Because the sample size is inconsistent and the statistical details are incomplete we can't draw a firm conclusion yet.

Having assessment criteria may help set better expectations and lead to more consistent grading of code.. How well this works depends on how good the rubric is and how consistently it's used. Future research should check these findings using the data report full statistical results and test grading reliability across different graders and tasks. For now the study's conclusions should stay limited, to the evidence that has been verified.

References

- [1]. D. D. Stevens and A. J. Levi, Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning, 2nd ed. Sterling, VA: Stylus Publishing, 2013.
- [2]. A. Jonsson and G. Svingby, "The use of scoring rubrics: Reliability, validity and educational consequences," Educational Research Review, vol. 2, no. 2, pp. 130–144, 2007.

- [3]. Y. M. Reddy and H. Andrade, "A review of rubric use in higher education," *Assessment & Evaluation in Higher Education*, vol. 35, no. 4, pp. 435–448, 2010.
- [4]. M. Ahmadzadeh, A. Elliman, and C. Higgins, "An analysis of patterns of debugging among novice computer science students," *ACM SIGCSE Bulletin*, vol. 37, no. 3, pp. 84–88, 2005.
- [5]. A. Ithantola et al., "Automatic assessment of programming assignments," *Proc. ACM Conf.*, pp. 123–130, 2010