Journal of Information Technology Education: Innovations in Practice

An Official Publication of the Informing Science Institute InformingScience.org

JITEiip.org

Volume 21, 2022

# Objective Assessment in Java Programming Language Using Rubrics

| | | |
|---|---|---|
| Priyanka Gupta* | AIIT, Amity University Uttar Pradesh, India | pjain2004@gmail.com |
| Deepti Mehrotra | ASET, Amity University Uttar Pradesh, India | dmehrotra@amity.edu |

*Corresponding author

## Abstract

| | |
|---|---|
| Aim/Purpose | This paper focuses on designing and implementing the rubric for objective JAVA programming assessments. An unsupervised learning approach was used to group learners based on their performance in the results obtained from the rubric, reflecting their learning ability. |
| Background | Students' learning outcomes have been evaluated subjectively using a rubric for years. Subjective assessments are simple to construct yet inconsistent and biased to evaluate. Objective assessments are stable, reliable, and easy to conduct. However, they usually lack rubrics. |
| Methodology | In this study, a Top-Down assessment approach is followed, i.e., a rubric focused on the learning outcome of the subject is designed, and the proficiency of learners is judged by their performance in conducting the task given. A JAVA rubric is proposed based on the learning outcomes like syntactical, logical, conceptual, and advanced JAVA skills. A JAVA objective quiz (with multiple correct options) is prepared based on the rubric criteria, comprising five questions per criterion. The examination was conducted for 209 students (100 from the MCA course and 109 from B.Tech. course). The suggested rubric was used to compute the results. K-means clustering was applied to the results to classify the students according to their learning preferences and abilities. |
| Contribution | This work contributes to the field of rubric designing by creating an objective programming assessment and analyzing the learners' performance using machine learning techniques. It also facilitates a reliable feedback approach offering various possibilities in student learning analytics. |
| Findings | The designed rubric, partial scoring, and cluster analysis of the results help us to provide individual feedback and also, group the students based on their learning skills. Like on average, learners are good at remembering the syntax |

Accepting Editor Tharrenos Bratitsis │ Received: May 25, 2022 │ Revised: July 29, August 16, October 6, October 28, 2022 │ Accepted: November 22, 2022.
Cite as: Gupta, P., & Mehrotra, D. (2022). Objective assessment in java programming language using rubrics. *Journal of Information Technology Education: Innovations in Practice, 21,* 155-173. https://doi.org/10.28945/5040

| | |
|---|---|
| | and concepts, mediocre in logical and critical thinking, and need more practice in code optimization and designing applications. |
| Recommendations for Practitioners | The practical implications of this work include rubric designing for objective assessments and building an informative feedback process. Faculty can use this approach as an alternative assessment measure. They are the strong pillars of e-assessments and virtual learning platforms. |
| Recommendations for Researchers | This research presents a novel approach to rubric-based objective assessments. Thus, it provides a fresh perspective to the researchers promising enough opportunities in the current era of digital education. |
| Impact on Society | In order to accomplish the shared objective of reflective learning, the grading rubric and its accompanying analysis can be utilized by both instructors and students. As an instructional assessment tool, the rubric helps instructors to align their pedagogies with the students' learning levels and assists students in updating their learning paths based on the informative topic-wise scores generated with the help of the rubric. |
| Future Research | The designed rubric in this study can be extended to other programming languages and subjects. Further, an adaptable weighted rubric can be created to execute a flexible and reflective learning process. In addition, outcome-based learning can be achieved by measuring and analyzing student improvements after rubric evaluation. |



| | |
|---|---|
| Keywords | rubric, JAVA programming, objective assessments, subjective assessments, rubric based evaluation |


## INTRODUCTION

An assessment evaluation process demands a useful and reliable feedback mechanism for the student's improvement. For a long time, subjective and objective assessments have been used to evaluate student learning and assign them grades. These evaluations aim to build and encourage reflective learning by measuring specific competencies such as knowledge, skills, and attitude (Alenezi & Faisal, 2020).

A subjective assessment (constructed-response) consists of elaborated and opinionated responses, while an objective assessment (selective-response) involves picking up correct alternatives out of preceded options (Shaban, 2014). Subjective assessment includes time-consuming creation, exhausting evaluation, ineffective presentation of ideas, and incompatible grading schemes (White, 2019). Objective assessments are more reliable, quick, easy to grade, and unbiased (White, 2019). Digital marking seems apt for objective assessments considering only the correct answers need to be fed to the machine, which saves the evaluators' time and effort (Souali et al., 2011). This type of selection-based result measures learners' proficiency level and higher-order thinking skills, given the appropriate construction of objective quizzes and the inculcation of distracting options (Belshaw et al., 2020; Rauf & Sultana, 2021). Examples of objective assessments include Multiple Choice Questions (MCQs), True/False, and matching (Bible et al., 2008).

A rubric is often used to solve most issues in subjective assessments (Debattista, 2018). It is a consistent framework representing multidimensional guidelines for scoring student work with defined criteria that reflect broad learning targets. The rubric elements include assessment criteria, performance descriptions, and scoring scales (Chan & Ho, 2019). Rubrics are commonly used for subjective assessments because of their established evaluation criteria, saving time and effort compared to manual evaluation. The rubric also allows for consistent marking and thorough comments (English et al., 2022). A rubric serves as a roadmap for the teacher's expectations of the pupils, ensuring that they meet those criteria (Boettger, 2010). According to some students and educators, broad and predetermined criteria-based subjective rubrics are a barrier to learners' critical and autonomous thinking (Chan & Ho, 2019). But still, consistent and standardized grading mechanisms are becoming popular among instructors and teaching assistants.

The rubric occupies a unique position among the other assessment and instructional tools such as checklists, manual evaluation, peer reviews, and portfolios due to its reliability and consistent grading technique. Moreover, it is a blueprint for the students to know about the instructors' expectations in the assessment. It is handy for the evaluators to not only grade the learners but also identify weak and strong points in the intended subject, improving their instructional techniques.

Incorporating a rubric with the objective assessments is not being practiced much because there are no elaborated replies to be rated across rubric criteria. The objective assessments are automatically graded once the selected responses are matched with an answer key, either manually or digitally. But, a rubric is much more than merely an evaluation tool. It is a method to extract the knowledge levels of learners and provide constructive feedback to foster imperative improvements (Chowdhury, 2019). In this regard, the combination of objective evaluations and rubrics will let evaluators adopt a simple and reliable analysis method while also assisting learners in self-regulated learning and critical thinking. This work aims to recommend this concoction and propose a model for competent and rubric-based objective assessments.

Attraction toward E-learning and e-assessments among learners and instructors is enhanced manifold (Khan et al., 2021) because the education and learning industry is shifting rapidly towards technical and virtual methods (Almossa, 2021; Caprara & Caprara, 2022; Dhawan, 2020), in particular, it rose as a most viable solution during Covid 19 Pandemic. Computer programming has emerged as one of the essential skills in this period of digital literacy, as every industry becomes technologically proficient and current progressive inventions are dependent on computer software (Contreras & Siu, 2015), giving rise to the mass enrollment of learners in computer programming courses across the world. In this scenario, fair assessment and feedback-based evaluation are critical components. In this case, an assessment rubric ensures the standardization of evaluation activities and the formulation of learning objectives (Mustapha et al., 2016). A rubric is used mainly in subjective programming assessments so far (Lindeborg, 2019; Mustapha et al., 2016).

An electronic rubric (E-rubric) is appropriate in this digitization period of the teaching-learning process (Subekti et al., 2021). Such a goal necessitates a consistent and rapid feedback-based quality learning mechanism to assess knowledge and cognition. If students receive comprehensive and timely feedback, they can more effectively focus on their strengths and weaknesses, stay motivated, and learn at their own speed.

This study emphasizes the potential of rubrics to evaluate objective programming evaluations by carefully and attentively picking questions that correspond to the predetermined criteria in the intended Java rubric, allowing the students' work to be assessed on each criterion. The analysis of such rubric-based objective assessments further may reveal the potential improvement scope for the learners and may suggest varied pedagogies and content-building ideas to the instructors. Therefore, this work analyses the rubric-based results using machine learning and suggests their practical implications. This study improves the transparency of evaluation elements and promotes the categorization of learners according to their learning needs and educational attainments. Instructors who get insights into students' learning and knowledge acquisition can choose an appropriate pedagogy to enhance the student's overall performance.

Usually, instructors conduct an examination and then prepare a rubric for evaluation. The primary limitation of this approach is designing an examination-specific rubric rather than a generic one. In such a scenario, the rubric is a mere evaluation tool. On the other hand, creating a generic rubric before the examination allows for designing rubric criteria-wise questions. In this manner, the same rubric may be utilized to prepare diverse assessments. This approach is referred to as the top-down assessment approach, which is followed in this study. Hence we first prepared the rubrics and then the questionnaire was designed. This approach ensures the formulations of the questions per the rubric's criteria, resulting in the reasonable prediction of the learning parameters of learners.

The rest of the paper is structured as follows: Section 2 presents a literature review mentioning the related studies, section 3 describes the research methodology, section 4 explains the proposed programming rubric, section 5 describes the cluster analysis of the rubric results, followed by section 6 which finally concludes the paper.

## LITERATURE REVIEW

An assessment is considered one of the critical components of higher education. It is an integral component of the teaching-learning process that is used for evaluating students' accomplishments (Kinash et al., 2018). A rubric is a well-adopted tool for evaluation and grading as it identifies the various criteria relevant to attainments of learning outcome and explicitly determines the possible levels of achievement of learner poor to excellent for given criteria (Jubaedah et al., 2020; Mrangu, 2022; Nsabayezu, Iyamuremye, et al., 2022). It is frequently used in education because it empowers competency-based evaluation and reflective learning methods (Velasco-Martínez & Tójar-Hurtado, 2018). Several rubrics are currently available to assist in the transition from mark-based to outcome-based learning. Numerous researchers, academic fraternity, and practitioners are involved in designing, applying, and validating rubrics to advance the learning mechanism. For instance, Ana et al. (2020) applied an e-rubric as a performance assessment instrument in Vocational Education to evaluate the learning process. They concluded that using e-rubric facilitated lecturers in measuring their students' skills in the practical curriculum. Salazar-Torres et al. (2021) proposed a rubric as an assessment tool for solving physical and mathematical problems. The results of this study indicated that the rubric provides extensive monitoring of student learning, accurate and timely feedback, and formative evaluation as a learning opportunity for students and teachers in physics and math. Nsabayezu, Mukiza, et al. (2022) explored the use of assessment rubrics in technology-based learning projects of organic chemistry. The results showed that the rubric-based assessment approach supported student learning, and instructors may quickly grade students' work and diagnose the students' strengths and weaknesses by providing formative feedback. Sonmez (2019) designed an assessment tool consisting of rubrics to evaluate activities related to verbal communication skills. The study suggested that the teachers require efficient strategies and practical assessment tools such as a rubric for designing the teaching process and material. Mrangu (2022) conducted a comprehensive literature review on rubric usage as an assessment tool in educational and program evaluations. The review found that the rubric is implemented in many studies and can be used for evaluation and instructional purposes. Teachers can identify student deficiencies and enhance instruction by using rubrics.

However, it is observed that most of the research and practice work in the rubric context is oriented toward subjective assessments. Virk et al. (2020) highlighted the importance of using rubrics in competency-based subjective assessment. According to this study, rubrics provide a learner-centered assessment technique that encourages behavioral change in learners while boosting the value and power of subjective assessment. Zedelius et al. (2019) tested an evaluation rubric to assess creative writing and stated that the rubric is only a viable alternative to subjective evaluation methods if it is based on objective textual features. According to Ab Rahman et al. (2020), the subjectivity of practical assessments causes biasing and, therefore, is difficult to measure without a rubric. They demonstrated that using a rubric scoring scale is appropriate for assessing the practical competence of students since it may translate qualitative criteria into quantitative forms on a grading scale. Most rubric development and implementation research focus on subjective assessments only (Grainger, 2021; Minnich et al., 2018; Schuller et al., 2019; Stanley et al., 2020).

A similar practice is pursued in the evaluation of computer programming assessments, where rubrics are used to evaluate subjective programming tests, and students are required to produce codes and write down concepts to receive grades. For instance, McGee et al. (2019) developed the scoring rubrics for the computer programming subjective assessment tasks. Chen et al. (2020) also developed a 7-point scoring rubric for assessing student explanations of programming codes in plain English, a subjective assessment. This study prepared the rubric based on the three dimensions of student answer quality: correctness, level of abstraction, and ambiguity. They demonstrated that a scoring rubric constructed using these three criteria could be reliable, align with the experienced instructors' intuition, and correctly connect to code writing competence. Von Wangenheim et al. (2018) employed a rubric in CodeMaster (a free web application tool to automatically assess block-based programs) to score computational reasoning based on static code analysis. According to the study, the tool can help students enhance their programming skills. Teachers can use it to assess entire classes, reducing their effort. Mustapha et al. (2016) focused on the grading inconsistencies in the programming assignments and proposed an assessment rubric for computer programming courses' cognitive, psychomotor, and affective domains. They implemented the rubrics and observed, using interrater reliability analysis, that the grades were consistent among the different instructors and the reliability of the developed rubric was also very good. Basu (2019) introduced a comprehensive multidimensional rubric that incorporates the evaluation of front-end project design with back-end sophistication of coding elements for assessing open-ended Block-Based Programming projects. Student project evaluation criteria are divided into overall competence, user experience (design elements), and coding and computer science constructs. Grover et al. (2018) proposed an extensive rubric to assess the programming projects built using block-based programming environments such as Scratch and App Inventor. The rubrics assessed student work along five dimensions: general considerations, design mechanics, user experience, fundamental coding constructs, and advanced coding constructs. In addition, Cateté et al. (2016) created an analytic rubric to help the graders of computer science to rate the programs written by students. The rubric criteria selected by this study are Accuracy, Efficiency, Reasoning, and Readability. Coenraad et al. (2021) discussed the utility of a structured rubric for evaluating curricular material in computer science. Their rubric was based on three primary criteria: Teacher accessibility, Equity, and Content. Eugene et al. (2016) and Lindeborg (2019) also demonstrated in their work that the use of a rubric for assessing computer programs proved to be beneficial for teachers and learners as the rubric can quickly test the critical and logical thinking of learners, which is a pivotal component in programming assessments.

The criteria selected for the rubric creation in the previous studies to assess programming skills are related to cognitive abilities and code writing skills, which is possible only in the case of subjective programming assessments. Therefore, creating a rubric for objective assessments such as MCQs demands a different criteria selection process that can precisely analyze the student learning outcomes by only the choices of ticked options. This research focuses on this aspect.

The concept of using multiple-choice or multiple-response assessment in computer programming is not new and is supported by several researchers. For example, Kuechler and Simkin (2003) highlighted the importance of multiple-choice tests over constructed-response ones in terms of their specific properties: automated mass grading, easy statistical analysis of results, the compensation of writing skills with the recollection of concepts, compatibility with the web-based courses, and consistent grading without a teacher's interference. Roberts (2006), as well as Simkin and Kuechler (2005) also mentioned and validated the relevance of objective assessments. Grover (2020) designed an assessment containing multiple-choice and open-response questions as a quality measure of student learning of basic programming concepts. They implemented the rubric only for the open-response item types and, therefore, understated the rubric usage in multiple-choice tests.

Research and practices on rubric usage in objective programming assessments are scarce, especially in multiple-choice or multiple-response examinations. Thus, this work is an attempt to involve rubrics in the objective assessments and reveal novel possibilities in this area.

## RESEARCH METHODOLOGY

Mixed-method research approach uses both qualitative and quantitative methods chronologically and then integrates them to summarize the research findings in the same study (Creswell & Clark, 2017). This work follows an exploratory sequential mixed-method research design. In this two-phase design, the qualitative data is collected, analyzed, and summarized, and then the quantitative data is gathered and examined to validate the qualitative findings (Guest & Fleming, 2015). Table 1 presents the phase-wise strategy followed in this research.

**Table 1: Exploratory sequential mixed-method research steps followed in this work**

| S/No. | Phase | Procedure |
|---|---|---|
| 1. | Qualitative data collection | Programming rubrics designed by various researchers were collected and explored |
| 2. | Qualitative data analysis | A Java rubric for objective assessment was developed after studying the previous rubrics and their gaps |
| 3. | Quantitative data collection | Responses of 209 students of MCA and B.Tech. to the Java multiple-response questionnaire were prepared in line with the seven criteria of the rubrics designed |
| 4. | Quantitative data analysis | Cluster analysis of the student results obtained from the Java questionnaire |
| 5. | Integration of Qualitative & Quantitative results | Descriptive statistics on the clusters formed to analyze and interpret the rubric-referenced scores |

The primary objective of this research study is to design the rubric for objective evaluation of JAVA programming assignments and assort the results using k-means clustering. To achieve this objective, the steps explained in Table 1 are followed. In the first phase of Qualitative data collection, the programming rubrics suggested by other researchers and practitioners were thoroughly studied and analyzed. Almost all the rubrics were designed keeping in mind the subjective programming assessments. But in this study, the focus is on objective programming assessments, and therefore, a new rubric was prepared in the second phase, Qualitative data analysis. The designed rubric was based on the seven criteria - Theory & Concepts, Syntax Knowledge, Conceptual Thinking & Skills, Critical Thinking, Logic Building & Thinking, Optimization Skills & Complexity, and Applications Design. The five-point scoring scale was used. In order to utilize this rubric, a JAVA multiple response-based quiz of 35 questions (5 questions each for seven criteria of the rubric) was prepared and tested on 209 students in the third phase, Quantitative data collection, where 100 students were from the MCA course and 109 were from B.Tech. (CSE) course. The test consisted of multiple correct answers, and students were unaware of the criteria-based question formations. In the next phase, Quantitative data analysis, the results of the students' tests were analyzed and scored according to the rubrics prepared. A powerful data mining approach, K-means clustering, was applied to the rubric referenced results. The rubric scores were communicated to the students as topic-wise feedback and further analysis on clusters formed was interpreted in the last phase, Integration of Qualitative and Quantitative results, to categorize the students in multiple learning groups based on their proficiency levels and application skills in all seven criteria.

## RUBRIC EXPLAINED

The creation of a rubric urges proper criteria selection (benchmarks for performance evaluation), scoring strategy (ratings on various levels), and performance descriptors (clarifications for the marking scheme) (Lee & Cherner, 2015).

A rubric for JAVA is developed in this study. However, this generic rubric can be used for any programming language with slight modifications. This rubric consists of seven critical measuring criteria to assess any object-oriented programming skills. The selected seven criteria is based on Bloom's

Taxonomy of learning domains. Bloom's Taxonomy is the ordering of cognitive skills. It shows how assessments can be designed concerning six levels: remember, understand, apply, analyze, evaluate and create (Chandio et al., 2016; Eber & Parker, 2007; Krathwohl, 2002).

The seven criteria of the rubric are listed in Table 2, along with their testing abilities, types of questions, and corresponding Bloom's Taxonomy learning level suitably:

**Table 2: Rubric with the corresponding Bloom's Taxonomy level**

| **Criteria** | **Testing parameter** | **Types of questions** | **0% (0)** No knowledge / Poor Knowledge | **25% (1-5)** Novice (Limited knowledge) | **50% (6-10)** Fluent (Needs more practice) | **75% (11-15)** Proficient (Good Programming skills) | **100% (16-20)** Expert (Well-versed) (Extensive knowledge of programming) | **Corresponding Bloom's Taxonomy Learning Level** |
|---|---|---|---|---|---|---|---|---|
| Theory & Concepts | Theoretical and Concept-based knowledge | Simple theory and concept learning questions | | | | | | Remember |
| Syntax Knowledge | Syntax knowledge | Small fragments of the codes testing the syntax checking | | | | | | Remember |
| Conceptual Thinking & Skills | Basic Conceptual understanding of the core programming concepts | Small output-based questions comprising only basic programming skills | | | | | | Understand |
| Critical Thinking | Application skills of various programming concepts together | Output-based programming questions combining some concepts | | | | | | Apply |
| Logic Building & Thinking | Deep thinking and analyzing skills of programming concepts | Tricky and compilation-based programming questions | | | | | | Analyze |
| Optimization Skills & Complexity | Code optimization skills and relative complexity of programs | Questions related to comparative performance and complexity of programs | | | | | | Evaluate |
| Applications Design | Software development skills in Java | Application-based questions | | | | | | Create |

The first two criteria, Theory & Concepts and Syntax Knowledge, test the remembering skill of the students (lowest learning level of Bloom's Taxonomy). The third criterion, Conceptual Thinking &

Skills, tests understanding basic programming concepts such as variables, data types, if-else, branching, and loops. The fourth criterion, Critical Thinking, tests applying skills as this criterion consists of output-based questions linking various concepts, such as looping and branching. The fifth criterion tests students' analyzing capabilities as this criterion contains tricky compilation-based questions. The sixth and seventh criteria, Optimization Skills & Complexity and Applications Design are related to the advanced Java concepts (according to the curriculum). The sixth criterion tests the evaluating skills of students as this criterion assesses the learners' knowledge and competence in optimizing the code and comparative performance. The seventh criterion tests the creating skills of the students as this criterion is associated with the software application building and deploying process. In this manner, the proposed rubric utilizes Bloom's Taxonomy in scrutinizing objective programming assessments. All the mentioned criteria explore the student learning and intellect around the principal dimensions of the JAVA programming language.

In the rubric, a five-point scoring scale was used for every criterion. Five questions per criterion were chosen to assess the skills related to that criterion. The examination consisted of 35 questions owing to the seven criteria. Each question was of four points. Because each question may have multiple correct answers, the correctness of each option was taken into account while calculating the final score. For example, if one question has b and c options correct, then the following rules are considered per option for marking the question –

a. option marked: 0 points otherwise 1 point
b. option marked: 1 point otherwise 0 points
c. option marked: 1 point otherwise 0 points
d. option marked: 0 points otherwise 1 point

Then, points for each option are added together to calculate the final score of the question. For instance, if one student marks only the b option, the total points are 3 (1+1+0+1), which means that his/her answer is 75% correct. This partial marking scheme extracts learners' proficiency level in each criterion as suggested in the rubric.

For each criterion, five questions are chosen, each of which is 4 points. Hence each criterion is of 20 points. The scoring range is according to the distribution of 20 points per criterion as follows (as per the rubric described in Table 2):

**0 points:** 0% correctness for that criterion interpreted as No/Very Poor knowledge of the criterion.

**1-5 points:** 25% correctness for that criterion interpreted as limited knowledge of the criterion.

**6-10 points:** 50% correctness for that criterion interpreted as fluency in the criterion.

**11-15 points:** 75% correctness for that criterion interpreted as proficiency in the criterion.

**16-20 points:** 100% correctness for that criterion interpreted as expertness in the criterion.

The learning levels of students per criterion have been computed using the manner described above. For instance, if a student attains 3 points in a particular criterion, he/she is a novice learner having minimal knowledge of that criterion. They can be advised to study these criteria topics thoroughly. Similarly, if a student attains 18 points in a particular criterion, he/she is an expert in that criterion. Therefore, he/she can be suggested to focus more on other programming competencies. This individual and specific feedback technique assist learners in getting insights into their expertise, moderate and poor programming aptitudes and improving them accordingly.

Table 3 shows the results of the first ten students of the data calculated according to the partial marking scheme discussed above. Table 4 shows the corresponding correctness of results according to the rubric, referred to as rubric scores in this work.

**Table 3: Scores of sample of 10 students as per the partial marking scheme**

| S/No. | Theory and Concepts | Syntax Knowledge | Conceptual Thinking & Skills | Critical Thinking | Logic Building & Thinking | Optimization Skills & Complexity | Applications Design |
|---|---|---|---|---|---|---|---|
| 1 | 11 | 19 | 13 | 17 | 12 | 10 | 9 |
| 2 | 19 | 16 | 18 | 14 | 13 | 15 | 18 |
| 3 | 10 | 11 | 15 | 8 | 17 | 13 | 8 |
| 4 | 5 | 16 | 11 | 16 | 11 | 13 | 11 |
| 5 | 7 | 12 | 13 | 9 | 13 | 13 | 10 |
| 6 | 17 | 20 | 20 | 16 | 13 | 11 | 15 |
| 7 | 11 | 15 | 13 | 17 | 8 | 7 | 8 |
| 8 | 16 | 18 | 13 | 16 | 10 | 9 | 9 |
| 9 | 19 | 16 | 18 | 18 | 19 | 14 | 9 |
| 10 | 12 | 17 | 19 | 11 | 14 | 12 | 9 |

**Table 4: Corresponding correctness of sample of 10 students as per the rubric described (Rubric Scores)**

| S/No. | Theory and Concepts | Syntax Knowledge | Conceptual Thinking & Skills | Critical Thinking | Logic Building & Thinking | Optimization Skills & Complexity | Applications Design |
|---|---|---|---|---|---|---|---|
| 1 | 75 | 100 | 75 | 100 | 75 | 50 | 50 |
| 2 | 100 | 100 | 100 | 75 | 75 | 75 | 100 |
| 3 | 50 | 75 | 75 | 50 | 100 | 75 | 50 |
| 4 | 25 | 100 | 75 | 100 | 75 | 75 | 75 |
| 5 | 50 | 75 | 75 | 50 | 75 | 75 | 50 |
| 6 | 100 | 100 | 100 | 100 | 75 | 75 | 75 |
| 7 | 75 | 75 | 75 | 100 | 50 | 50 | 50 |
| 8 | 100 | 100 | 75 | 100 | 50 | 50 | 50 |
| 9 | 100 | 100 | 100 | 100 | 100 | 75 | 50 |
| 10 | 75 | 100 | 100 | 75 | 75 | 75 | 50 |

The result presented in Table 4 is interpreted in terms of partial correctness. For instance, the first student achieves 75% correctness in the first criterion, 100% correctness in the second criterion, and so on. These rubric scores are communicated to the students as feedback. From these scores, the learners get an idea of their learning levels in the mentioned criteria of Java programming. This type of information helps them to prioritize their learning paths accordingly.

The average of each criterion results are as follows in Table 5:

**Table 5: Average of criteria-wise rubric referenced results**

| Criteria | Average |
|---|---|
| Theory and Concepts | 75.83732 |
| Syntax Knowledge | 84.33014 |
| Conceptual Thinking & Skills | 79.90431 |
| Critical Thinking | 76.07656 |
| Logic Building & Thinking | 76.19617 |
| Optimization Skills & Complexity | 68.0622 |
| Applications Design | 59.33014 |

Table 5 shows that overall the students have more syntactic knowledge than any other criterion on average. The minimum score is achieved in the Applications Design criterion. In addition, they are mediocre in all other areas. It explains that the learners are more comfortable memorizing the concepts and solving short output-based questions but are comparatively less inclined toward code optimization and designing the applications. Thus, they must be encouraged to focus more on advanced programming paradigms and application building in real-world situations. This assessment form encourages students to discover and improve their weakest learning areas.

Although this analysis is sufficient for individual feedback in objective assessments, a clustering approach further helps segregate the learners as per their learning competencies, which will assist instructors in knowing about their class. The students would be divided into separate groups/clusters where each cluster is unique regarding their learning preferences. This information is helpful for pedagogy and content design per the class's needs. For this purpose, clustering is applied to the above-calculated results and is described in the next section.

## CLUSTER ANALYSIS

Clustering is an unsupervised machine learning algorithm in pattern analysis. Its primary purpose is to identify the dissimilar data chunks or clusters where each cluster contains similar data items (Rodriguez et al., 2019). This data analysis approach provides valuable insights into the relevant data aspects. However, there are various clustering algorithms available, but this study chooses the most popular K-means partitioning clustering algorithm because of its simplicity, efficiency, and lower execution time (Karthikeyan & Aruna, 2013).

### *FINDING THE OPTIMUM NUMBER OF CLUSTERS*

The k-means algorithm aims to minimize the sum of all intra-cluster distances and maximize the sum of all inter-cluster distances (Singh & Gill, 2013) to ensure the distinctiveness of all clusters, which is the essential requirement of a clustering algorithm. Choosing the appropriate number of clusters is essential. This study uses the WSS plot and NbClust() functions to find the optimal number of clusters. WSS plot is an elbow method displaying an elbow-shaped curve having a number of clusters on the X-axis and within-groups sums of squares on the Y-axis. The kink or bend on the curve suggests the appropriate number of clusters (Maheswari, 2019). WSS plot for the students' rubric results evaluated in this work is shown in Figure 1.

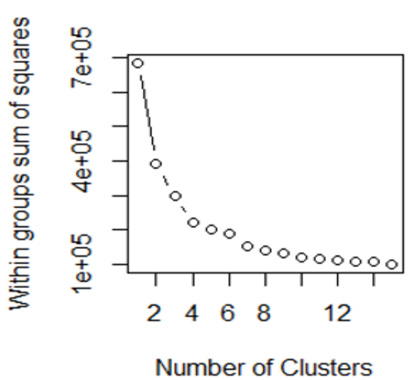


**Figure 1: WSS plot for the optimal number of clusters**

The disadvantage of the elbow method is its ambiguity (Maheswari, 2019). As visible in Figure 1, there is not a sharp bend in the suggested number of clusters. Thus, only this method is insufficient to calculate the optimal number of clusters. As a result, the NbClust library and functions are being used in this research, which provides the possible number of clusters after running experiments such as the Hubert index and D index. These two graphical methods for determining the best number of clusters are shown in Figures 2 and 3. The significant peak values in both procedures correspond to the desired number of clusters.

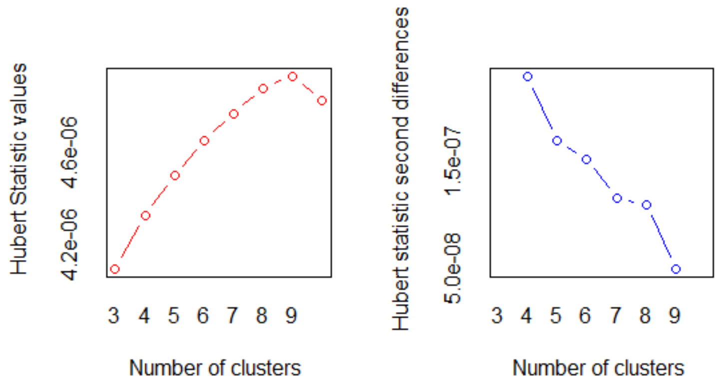


**Figure 2: Hubert index method for the optimal number of clusters**

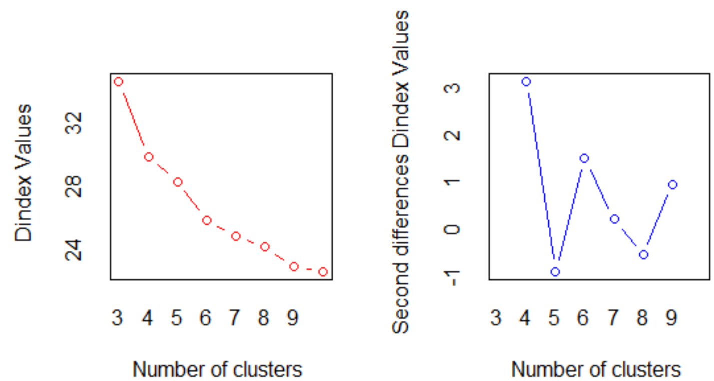


**Figure 3: Dindex method for the optimal number of clusters**

The above two methods indicate four as the optimal number of clusters. Additionally, the NbClust package suggests the optimal number of clusters by choosing several criteria, as depicted in Figure 4.

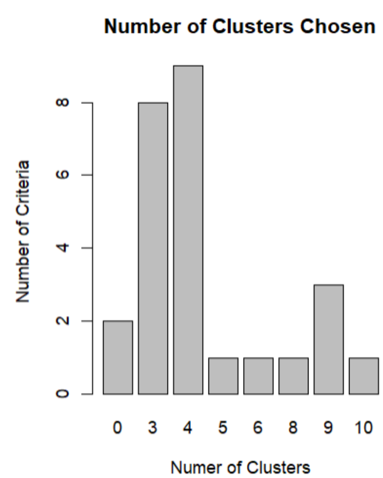


**Figure 4: No. of Clusters suggested by various criteria provided by the NbClust package**

In the output of NbClust, among all indices in fig.4, 8 proposed three as the best number of clusters, 9 proposed four as the best number of clusters, 1 proposed five as the best number of clusters, 1 proposed six as the best number of clusters, 1 proposed eight as the best number of clusters, 3 proposed nine as the best number of clusters and 1 proposed ten as the best number of clusters. As a result, the best number of clusters, according to the majority rule, is four.

## *Applying K-Means Algorithm*

After calculating and verifying the best number of clusters as four, the k-means clustering algorithm was applied to the students' Java test results. These results were evaluated using the rubric designed which are referred to as "Rubric Scores" in Table 4. Figure 5 displays a cluster plot of four clusters as a result. The cluster centers or aggregate values are shown in Table 6.

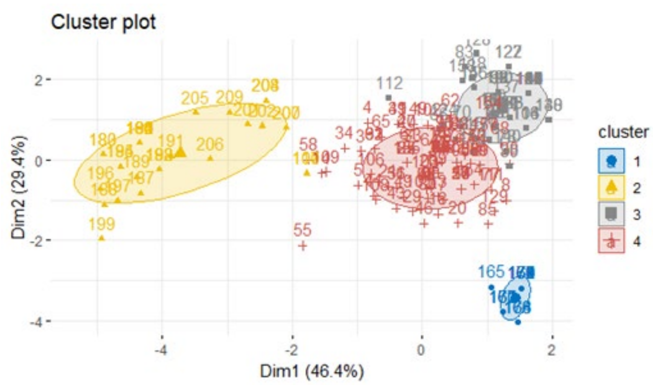


**Figure 5: Cluster Plots with k=4**

**Table 6: Cluster centers of four clusters generated**

| Cluster No. | Theory and Concepts | Syntax Knowledge | Conceptual Thinking & Skills | Critical Thinking | Logic Building & Thinking | Optimization Skills & Complexity | Applications Design |
|---|---|---|---|---|---|---|---|
| 1 | 98.75000 | 100.0000 | 75.00000 | 45.0000 | 46.25000 | 25.00000 | 25.00000 |
| 2 | 26.56250 | 32.81250 | 43.75000 | 60.9375 | 78.90625 | 84.37500 | 87.50000 |
| 3 | 96.48438 | 97.65625 | 98.04688 | 94.5312 | 89.45312 | 76.95312 | 57.42188 |
| 4 | 73.65591 | 89.51613 | 80.91398 | 75.2688 | 72.58065 | 65.59140 | 58.33333 |

## *Interpretation of Clustering Results*

The quality and correctness of the clustering depend on the distinctiveness of the clusters. K-means method allocates every item to only one cluster nearest to it in terms of a measure of Euclidean distance (Oyelade et al., 2010). Each cluster must be unique. In this work, these requirements are verified in Figure 5 and Table 6. In Fig.5, it is visible that all the four clusters are discrete, and no overlapping is there. Table 6 represents the cluster centers of the clusters generated. The cluster center of a particular cluster is the aggregate value around which the other values of that cluster persist. Thus, according to Table 6, cluster 1 is the group of students having around 98.75 rubric scores in Theory and Concepts, 100 rubric scores in Syntax Knowledge, 75 rubric scores in Conceptual Thinking & Skills, and so on.

Similarly, there are the cluster center values for the other three clusters. Figure 6 presents the line chart of Table 6, where all the seven criteria are on the X-axis. Cluster 1 center values decrease rapidly criteria-wise; Cluster 2 center values increase while cluster 3 and cluster 4 center values show a moderate decrease. Each cluster is distinct and exhibits diverse characteristics in terms of student achievement in the seven programming criteria.

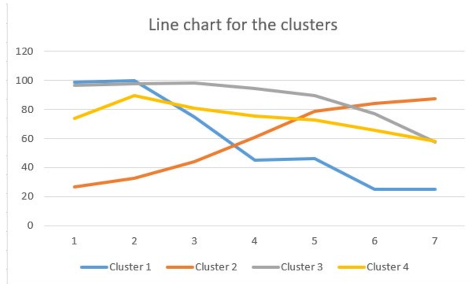


**Figure 6: Line Chart for the cluster center values**

## *Analysis of Results*

The four clusters engendered in this work, as shown in Table 6, segregate the students into four distinct batches of sizes exhibited in Table 7. Cluster 1 is a group of students who scores well in Theory and Concepts, Syntax Knowledge, and Conceptual Thinking & Skills; moderate in Critical Thinking and Logic Building & Thinking; and less in Optimization Skills & Complexity and Applications Design. Cluster 2 is a group of students who scores less in Theory and Concepts and Syntax Knowledge; moderate in Conceptual Thinking & Skills and Critical Thinking; and good in Logic Building & Thinking, Optimization Skills & Complexity, and Applications Design. Cluster 3 is a group of students who score very well in Theory and Concepts, Syntax Knowledge, Conceptual Thinking & Skills, Critical Thinking, and Logic Building & Thinking but also score well in Optimization Skills & Complexity and moderate in Applications Design. Cluster 4 is a group of students who scores moderately in every criterion but good in Syntax Knowledge.

Consequently, the students of cluster 1 have only basic knowledge of Java programming. These pupils can recall conceptual theory and syntax, but their application skills are ordinary, requiring more practice. They are weak in optimizing the code and building the applications in real-life scenarios. As a result, these learners can be recommended to concentrate more on implementing and analyzing what they have learned. Similarly, cluster 2 students are better at applying skills critically and rationally, but they struggle to recall theoretical concepts and syntactic knowledge. These students are excellent at optimizing the code and building real-life applications. This group is more interested in application design and reluctant to recall the concepts. As a result, this group can apply and analyze but cannot remember the concepts easily. Cluster 3 is a collection of students who are very good at remembering the concepts and syntax. They can also analyze critically and logically, but they are average in code optimization and application development. This batch can memorize the syntax and theory and analyze the learned concepts but needs to focus a little more on optimization and application design. Cluster 4 is a group of average students in all categories. As suggested to cluster 3 students, this group is also advised to pay a little more attention to the application building and code optimization skills, but this batch is advised to pay heed to the other criteria, too, unlike the cluster 3 batch. As a result, this clustering method identifies learners' poor, average, above-average, and strong programming domains. These clusters are communicated to the instructors so that they divide their students into groups based on their knowledge levels and prepare their lessons accordingly.

**Table 7: No. of students per cluster**

| Cluster No. | Size (No. of students) |
|---|---|
| 1 | 20 |
| 2 | 32 |
| 3 | 64 |
| 4 | 93 |

In addition, it can be seen from Table 7 that cluster 1 is the smallest cluster having only 20 students, and cluster 4 is the largest cluster having 93 students. This observation implies that a considerably lower number of students in our dataset excel at remembering syntax and concepts but struggle in logic building, critical thinking, code optimization, and application design. A considerable number of students are average in practically every aspect. This analysis gives a sense of a group's preferences for various programming categories.

## CONCLUSION AND FUTURE SCOPE

This paper focuses on developing rubrics in a top-down manner for JAVA objective assessments that include multiple response questions. Creating these objective assessments is simple and quick (Patra & Saha, 2019). Furthermore, they facilitate consistent and objective mass evaluation, a prerequisite of today's mob learning platforms. If the questions are carefully selected, they can provide competent information regarding student learning. It is proved in the existing research that rubric-based assessment improves student learning (Miknis et al., 2020), but no work acclaimed the significance of the use of rubrics in objective assessments.

In this work, a rubric is prepared in which seven criteria were selected to assess the students' responses. The criteria chosen are in agreement with Bloom's taxonomy. A JAVA Multiple Response Questionnaire is created, with five questions chosen for each criterion to assess students' learning and comprehension related to that criterion. It was conducted on 209 students of MCA and B.Tech. courses. The rubric scores (shown in Table 4) were calculated using the partial scoring technique described in the rubric section. These rubric scores suggest the problem areas for the learners where they need to concentrate more to perform better. This approach ensures skill-wise feedback to each student, and it is better than just knowing the total number of correct answers or the total score. Clustering is also applied to the results to categorize the learners according to their expertise in different programming criteria. This rubric-based assessment aids the instructors in developing the subsequent curriculum, pedagogy, and class content.

Conclusively, some students are good at remembering, comprehending, and applying programming principles but struggle with analyzing, evaluating, and developing domains, whereas others excel at applying, evaluating, and analyzing but are weak in remembering and understanding programming concepts. This separation also assists industrial enterprises in employing students who meet their requirements.

The whole idea behind every type of assessment is to measure the student learning outcomes in the end. An outcome is generally referred to as the knowledge or skills acquired by the student throughout the course and can be measured after a certain time period (Seybert, 2002). This work focuses on proposing and analyzing a rubric for objective assessments. The rubric scores are communicated to the learners which suggests topic-wise improvements but the outcome measurement of student achievements and rubric evaluation are out of the scope of this work and may be included in the future study in this area.

Additionally, the students examined in this work are from two separate courses: B.Tech. and MCA. They are regarded as programming learners in this work, but future research may undertake a differential analysis of the students in the two courses, yielding a course-wise analysis.

The other future scopes of this work are evaluating the rubric's reliability and validity, developing and implementing it for objective assessments in other subjects, and sharing and discussing the rubric with students prior to the assessment so that they are aware of the criteria on which they will be evaluated. The rubric design can also be extended to adaptable rubrics that offer detailed sub-criteria for each programming criteria if demanded (Agost et al., 2018), accommodating diverse learning methods and speeds.

## REFERENCES


Ab Rahman, A., Hanafi, N. M., Yusof, Y., Mukhtar, M. I., Awang, H., & Yusof, A. M. (2020). The effect of rubric on rater's severity and bias in TVET laboratory practice assessment: Analysis using many-facet Rasch measurement. *Journal of Technical Education and Training*, *12*(1), 57–67. https://penerbit.uthm.edu.my/ojs/index.php/JTET/article/view/3248

Agost, M. J., Otey, J., Company, P., & Contero, M. (2018, November). New features for providing e-rubrics with adaptable properties. *Proceedings of the 11th Annual International Conference of Education, Research and Innovation (ICERI2018)* (pp. 1312-1316). Seville, Spain: International Academy of Technology, Education and Development (IATED). https://doi.org/10.21125/iceri.2018.0130

Alenezi, H. S., & Faisal, M. H. (2020). Utilizing crowdsourcing and machine learning in education: Literature review. *Education and Information Technologies*, *25*(4), 2971–2986. https://doi.org/10.1007/s10639-020-10102-w

Almossa, S. Y. (2021). University students' perspectives toward learning and assessment during COVID-19. *Education and Information Technologies*, *26*(6), 7163–7181. https://doi.org/10.1007/s10639-021-10554-8

Ana, A., Yulia, C., & Muktiarni, M. (2020). Electronic rubric: Evaluation tool in the assessment process in vocational education. *Journal of Engineering Science and Technology*, *15*(6), 3789-3802. http://jestec.taylors.edu.my/Vol%2015%20issue%206%20December%202020/15_6_20.pdf

Basu, S. (2019, February). Using rubrics integrating design and coding to assess middle school students' open-ended block-based programming projects. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE 2019)* (pp. 1211–1217). Minneapolis, MN: ACM. https://doi.org/10.1145/3287324.3287412

Belshaw, A., Mackie, A., Phillips, H. K., Rodrigues, H., & Russell, A. (2020). Comment on: Creating assessments as an active learning strategy: What are students' perceptions? A mixed methods study. *Medical Education Online*, *25*(1), Article: 1709277. https://doi.org/10.1080/10872981.2019.1709277

Bible, L., Simkin, M. G., & Kuechler, W. L. (2008). Using multiple-choice tests to evaluate students' understanding of accounting. *Accounting Education*, *17*(1), S55-S68. https://doi.org/10.1080/09639280802009249

Boettger, R. K. (2010). Rubric use in technical communication: Exploring the process of creating valid and reliable assessment tools. *IEEE Transactions on Professional Communication*, *53*(1), 4-17. https://doi.org/10.1109/TPC.2009.2038733

Caprara, L., & Caprara, C. (2022). Effects of virtual learning environments: A scoping review of literature. *Education and Information Technologies*, *27*(3), 3683–3722. https://doi.org/10.1007/s10639-021-10768-w

Cateté, V., Snider, E., & Barnes, T. (2016, July). Developing a rubric for a creative CS principles lab. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16)* (pp. 290–295). Arequipa, Peru: ACM. https://doi.org/10.1145/2899415.2899449

Chan, Z., & Ho, S. (2019). Good and bad practices in rubrics: The perspectives of students and educators. *Assessment & Evaluation in Higher Education*, *44*(4), 533-545. https://doi.org/10.1080/02602938.2018.1522528

Chandio, M. T., Pandhiani, S. M., & Iqbal, R. (2016). Bloom's taxonomy: Improving assessment and teaching-learning process. *Journal of Education and Educational Development*, *3*(2), 203-221. https://jmsnew.iobmresearch.com/index.php/joeed/article/download/194/330

Chen, B., Azad, S., Haldar, R., West, M., & Zilles, C. (2020, February). A validated scoring rubric for explain-in-plain-English questions. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)* (pp. 563-569). Portland, OR: ACM. https://doi.org/10.1145/3328778.3366879

Chowdhury, F. (2019). Application of rubrics in the classroom: A vital tool for improvement in assessment, feedback and learning. *International Education Studies*, *12*(1), 61-68. https://doi.org/10.5539/ies.v12n1p61

Coenraad, M., Hopcraft, C., Jozefowicz, J., Franklin, D., Palmer, J., & Weintrop, D. (2021). Helping teachers make equitable decisions: Effects of the TEC rubric on teachers' evaluations of a computing curriculum. *Computer Science Education*, *31*(3), 400-429. https://doi.org/10.1080/08993408.2020.1788862

Contreras, G. J., & Siu, K. W. M. (2015). Computer programming for all: A case-study in product design education. *Procedia – Social and Behavioral Sciences*, *182*, 388–394. https://doi.org/10.1016/j.sbspro.2015.04.799

Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*. SAGE Publications.

Debattista, M. (2018). A comprehensive rubric for instructional design in e-learning. *International Journal of Information and Learning Technology*, *35*(2), 93–104. https://doi.org/10.1108/IJILT-09-2017-0092

Dhawan, S. (2020). Online learning: A panacea in the time of COVID-19 crisis. *Journal of Educational Technology Systems*, *49*(1), 5–22. https://doi.org/10.1177/0047239520934018

Eber, P. A., & Parker, T. S. (2007). Assessing student learning: Applying Bloom's taxonomy. *Human Service Education*, *27*(1), 45-53. https://www.mindmeister.com/generic_files/get_file/763537

English, N., Robertson, P., Gillis, S., & Graham, L. (2022). Rubrics and formative assessment in K-12 education: A scoping review of literature. *International Journal of Educational Research*, *113*, 101964. https://doi.org/10.1016/j.ijer.2022.101964

Eugene, K., Stringfellow, C., & Halverson, R. (2016). The usefulness of rubrics in computer science. *Journal of Computing Sciences in Colleges*, *31*(4), 5–20. https://dl.acm.org/doi/10.5555/2904127.2904129

Grainger, P. (2021). Enhancing assessment literacies through development of quality rubrics using a triad based peer review process. *Journal of University Teaching & Learning Practice*, *18*(4), Article 4. https://doi.org/10.53761/1.18.4.4

Grover, S. (2020, February). Designing an assessment for introductory programming concepts in middle school computer science. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE'20)* (pp. 678-684). Portland, OR: ACM. https://doi.org/10.1145/3328778.3366896

Grover, S., Basu, S., & Schank, P. (2018, February). What we can learn about student learning from open-ended programming projects in middle school computer science. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE'18)* (pp. 999–1004). Baltimore, Maryland: ACM. https://doi.org/10.1145/3159450.3159522

Guest, G., & Fleming, P. (2015). Mixed methods research. In G. Guest & E. E. Namey (Eds.), *Public health research methods* (pp. 581-614). SAGE Publications. https://doi.org/10.4135/9781483398839

Jubaedah, Y., Yulia, C., Muktiarni, M., & Maosul, A. (2020). Usability testing electronic rubric of performance assessment. *Journal of Physics: Conference Series*, *1456*(1), 012016. https://doi.org/10.1088/1742-6596/1456/1/012016

Karthikeyan, M., & Aruna, P. (2013). Probability based document clustering and image clustering using content-based image retrieval. *Applied Soft Computing*, *13*(2), 959-966. https://doi.org/10.1016/j.asoc.2012.09.013

Khan, M. A., Vivek, Nabi, M. K., Khojah, M., & Tahir, M. (2021). Students' perception towards e-learning during Covid-19 pandemic in India: An empirical study. *Sustainability*, *13*(1), 57. https://doi.org/10.3390/su13010057

Kinash, S., McGillivray, L., & Crane, L. (2018). Do university students, alumni, educators and employers link assessment and graduate employability? *Higher Education Research & Development*, *37*(2), 301-315. https://doi.org/10.1080/07294360.2017.1370439

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice*, *41*(4), 212-218. https://doi.org/10.1207/s15430421tip4104_2

Kuechler, W. L., & Simkin, M. G. (2003). How well do multiple choice tests evaluate student understanding in computer programming classes? *Journal of Information Systems Education*, *14*(4), 389-400. http://jise.org/Volume14/n4/JISEv14n4p389.pdf

Lee, C. Y., & Cherner, T. S. (2015). A comprehensive evaluation rubric for assessing instructional apps. *Journal of Information Technology Education: Research*, *14*, 21-53. https://doi.org/10.28945/2097

Lindeborg, C. (2019). *Improved programming assignment assessment with grading rubrics* [Degree project. Stockholm, Sweden: KTH Royal Institute Of Technology, School of Electrical Engineering and Computer Science]. https://www.diva-portal.org/smash/get/diva2:1337083/FULLTEXT01.pdf

Maheswari, K. (2019). Finding best possible number of clusters using K-Means algorithm. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*(1S4), 533-538. https://doi.org/10.35940/ijeat.A1119.1291S419

McGee, S., Greenberg, R. I., McGee-Tekula, R., Duck, J., Rasmussen, A. M., Dettori, L., & Reed, D. F. (2019, February). An examination of the correlation of exploring computer science course performance and the development of programming expertise. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)* (pp. 1067-1073). Minneapolis, MN: ACM. https://doi.org/10.1145/3287324.3287415

Miknis, M., Davies, R., & Johnson, C. S. (2020). Using rubrics to improve the assessment lifecycle: A case study. *Higher Education Pedagogies*, *5*(1), 200–209. https://doi.org/10.1080/23752696.2020.1816843

Minnich, M., Kirkpatrick, A. J., Goodman, J. T., Whittaker, A., Chapple, H. S., Schoening, A. M., & Khanna, M. M. (2018). Writing across the curriculum: Reliability testing of a standardized rubric. *Journal of Nursing Education*, *57*(6), 366-370. https://doi.org/10.3928/01484834-20180522-08

Mrangu, L. (2022). Rubric as assessment tool for lecturers and students in higher education institution. *Acta Pedagogia Asiana*, *1(*1), 26-33. https://doi.org/10.53623/apga.v1i1.98

Mustapha, A., Samsudin, N. A., Arbaiy, N., Mohamed, R., & Hamid, I. R. (2016). Generic assessment rubrics for computer programming courses. *TOJET: The Turkish Online Journal of Educational Technology*, *15*(1), 53–68. https://files.eric.ed.gov/fulltext/EJ1086221.pdf

Nsabayezu, E., Iyamuremye, A., Nahimana, J. P., Mukiza, J., Kampire, E., & Nsengimana, T. (2022). The progress in the application of rubric materials in chemistry teaching and students' learning enhancement during 21st century: A systematic review. *Discover Education*, *1*(1), Article 5. https://doi.org/10.1007/s44217-022-00005-y

Nsabayezu, E., Mukiza, J., Iyamuremye, A., Mukamanzi, O. U., & Mbonyiryivuze, A. (2022). Rubric-based formative assessment to support students' learning of organic chemistry in the selected secondary schools in Rwanda: A technology-based learning. *Education and Information Technologies*, *27*(5), 12251–12271. https://doi.org/10.1007/s10639-022-11113-5

Oyelade, O. J., Oladipupo, O. O., & Obagbuwa, I. C. (2010). Application of K-Means clustering algorithm for prediction of students' academic performance. *International Journal of Computer Science and Information Security (IJCSIS)*, *7*(1), 292-295. https://arxiv.org/ftp/arxiv/papers/1002/1002.2425.pdf

Patra, R., & Saha, S. K. (2019). A hybrid approach for automatic generation of named entity distractors for multiple choice questions. *Education and Information Technologies*, *24*(2), 973–993. https://doi.org/10.1007/s10639-018-9814-3

Rauf, A. R., & Sultana, S. (2021). Effect of faculty training on quality of multiple choice questions. *Rawal Medical Journal*, *46*(2), 430–433. https://www.rmj.org.pk/fulltext/27-1599644704.pdf

Roberts, T. S. (2006, January). The use of multiple choice tests for formative and summative assessment. *Proceedings of the 8th Australasian Conference on Computing Education (ACE '06), 52* (pp. 175–180). Hobart, Australia: Australian Computer Society. https://dl.acm.org/doi/10.5555/1151869.1151892

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS ONE*, *14*(1), e0210236. https://doi.org/10.1371/journal.pone.0210236

Salazar-Torres, J., Leal, O. R., & Ortega, M. V. (2021, July). The rubric as an assessment tool for solving problem situations in the physics and mathematics teaching context. *Journal of Physics: Conference Series*, *1981*(1), 012018. IOP Publishing. https://doi.org/10.1088/1742-6596/1981/1/012018

Schuller, K. A., Cronin, C. E., Nicks, S. E., Jing, X., Kingori, C., & Morrone, M. (2019). Development and application of a rubric to compare strategies for improving access to health care in rural communities in the United States. *Evaluation and Program Planning*, *74*, 61-68. https://doi.org/10.1016/j.evalprogplan.2019.02.013

Seybert, J. A. (2002). Assessing student learning outcomes. *New Directions for Community Colleges*, *2002*(117), 55-66. https://doi.org/10.1002/cc.53

Shaban, A. M. S. (2014). A comparison between objective and subjective tests. *Journal of the College of Languages (JCL)*, (30), 44-52. https://jcolang.uobaghdad.edu.iq/index.php/JCL/article/download/169/109

Simkin, M. G., & Kuechler, W. L. (2005). Multiple-choice tests and student understanding: What is the connection? *Decision Sciences Journal of Innovative Education*, *3*(1), 73–98. https://doi.org/10.1111/j.1540-4609.2005.00053.x

Singh, S., & Gill, N. S. (2013). Analysis and study of K-Means clustering algorithm. *International Journal of Engineering Research & Technology (IJERT)*, *2*(7), 2546-2551. https://www.ijert.org/research/analysis-and-study-of-k-means-clustering-algorithm-IJERTV2IS70648.pdf

Sonmez, H. (2019). The strategies for designing activity related to listening/following skills and assessment rubric. *Educational Policy Analysis and Strategic Research*, *14*(2), 124-154. https://doi.org/10.29329/epasr.2019.201.7

Souali, K., El Afia, A., Faizi, R., & Chiheb, R. (2011, April). A new recommender system for e-learning environments. *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS 2011)* (pp. 1-4). Ouarzazate, Morocco: IEEE. https://doi.org/10.1109/ICMCS.2011.5945630

Stanley, D., Coman, S., Murdoch, D., & Stanley, K. (2020). Writing exceptional (specific, student and criterion-focused) rubrics for nursing studies. *Nurse Education in Practice*, *49*, 102851. https://doi.org/10.1016/j.nepr.2020.102851

Subekti, S., Ana, A., Muktiarni, M., & Dwiyanti, V. (2021). E-rubric to measure employability skills. *Journal of Engineering Science and Technology*, *16*(1), 851–860. http://jestec.taylors.edu.my/Vol%2016%20issue%201%20February%202021/16_1_60.pdf

Velasco-Martínez, L. C., & Tójar-Hurtado, J. C. (2018). Competency-based evaluation in higher education – Design and use of competence rubrics by university educators. *International Education Studies*, *11*(2), 118-132. https://doi.org/10.5539/ies.v11n2p118

Virk, A., Joshi, A., Mahajan, R., & Singh, T. (2020). The power of subjectivity in competency-based assessment. *Journal of Postgraduate Medicine*, *66*(4), 200-205. https://doi.org/10.4103/jpgm.JPGM_591_20

Von Wangenheim, C. G., Hauck, J. C. R., Demetrio, M. F., Pelle, R., da Cruz Alves, N., Barbosa, H., & Azevedo, L. F. (2018). CodeMaster – Automatic assessment and grading of app inventor and snap! programs. *Informatics in Education*, *17*(1), 117-150. https://doi.org/10.15388/infedu.2018.08

White, A. R. (2019). *A discussion and students' thoughts on the assessment methods used at the tertiary level*. HAL Open Science. https://hal.archives-ouvertes.fr/hal-01976361/document

Zedelius, C. M., Mills, C., & Schooler, J. W. (2019). Beyond subjective judgments: Predicting evaluations of creative writing from computational linguistic features. *Behavior Research Methods*, *51*(2), 879–894. https://doi.org/10.3758/s13428-018-1137-1

## AUTHORS

**Ms. Priyanka Gupta** is currently working in the IT department of Vivekananda Institute of Professional Studies as an Assistant Professor. She completed her MCA post-graduation from GGSIPU and B.Sc. (H) in Physics from Delhi University. She is pursuing a Ph.D. in Information Technologies from Amity University, Noida. She has also done a diploma in 'O' level from DOEACC. She is NET qualified faculty in computer science. She possesses over 9 years of experience in teaching computer science subjects such as C, C++, Java, Digital Electronics, Computer Architecture, Computer Graphics, Web Technologies, Multimedia, Data Warehousing & Data Mining, and Operating Systems. She is actively involved in research practices in computer science for 6 years and published many research papers in various conferences and journals. She has attended several Faculty Development Programs, Workshops, and Seminars for widening academic excellence and exploring new opportunities. Her research interest areas include Internet of Things, Data Mining, Smart Things, Rubrics, and Intelligent Tutoring Systems.

**Prof. Deepti Mehrotra** is Head of Accreditation, Ranking & Quality Assurance and Professor in IT Department of Amity University, Noida, Uttar Pradesh. She has 25 years of experience in teaching and research. Her research interests include data science, machine learning, blockchain, artificial intelligence, soft computing, and nature-inspired algorithms. She has been invited as an expert at forums like FDP and national/international conferences. 14 PhD scholars have submitted/ awarded degrees, and six are currently pursuing their research work under her guidance. She has published more than 175 research papers in SCI/Scopus indexed international journals/ international conferences/ book chapters. The publication includes Q1 journals and international collaborations. Her research work has been cited by more than 1500 researchers and has an H-index of 19. She has been associated with International Conferences as publication chair, TPC member, and reviewer of various international journals published by IEEE, Elsevier, Springer etc. She received Gold Medal in post-graduation from Lucknow University. She is involved in multiple accreditations & ranking processes and is a member of different academic Committees.