Exploring the Validity of Continuous Assessment in a First-Year Programming Course at a Comprehensive Open Distance e-Learning University

Dalize van Heerden and Mmafani Serote
University of South Africa, Pretoria, South Africa

vheerme1@unisa.ac.za

Abstract: Continuous assessments are commonly used to determine students' level of understanding to implement measures to address any shortcomings. Assessment practices of programming courses at distance universities have always had numerous drawbacks due to large student bodies and time constraints. The large student body, in most instances, creates questions related to the validity and authenticity of the assessments. Time constraints in grading large numbers of assessments create questions related to the validity and authenticity of the feedback on assessments. The assumption is that the more assessments required for programming courses in a distance learning environment, the more chances there are for cheating and the less time there is to provide feedback for learning. This paper will investigate and report on assessment practices that were adopted in a first-year programming course to ensure valid authentic continuous assessment. Experiences shared in this paper offer programming instructors and the e-learning community, in general, an opportunity to make decisions on the extent to which they can use the continuous assessment strategies employed in their own settings.

Keywords: Continuous Assessment, Validity, Authenticity, Programming, E-Learning

1. Introduction

Valid and authentic continuous assessment is necessary to track student progress, give prompt feedback, promote deep learning, and encourage active learning. Also, it is crucial to guarantee that students can successfully learn and apply the abilities and expertise required to succeed in the programming sector. Valid, authentic continuous assessment promotes critical thinking, guards against plagiarism, and helps students prepare for the workforce. This paper will explore the validity and authenticity of continuous assessment in a programming course at a Comprehensive Open Distance e-Learning (CODeL) university.

The paper is based on a first-year programming course at a CODeL university. The course allows students to demonstrate an understanding of the concepts of interface design in view of its importance to the success of a computer-based product as well as a clear understanding of problem statements as stated by computer users in various industries. The course teaches the students to identify and apply user interface design principles which allows them to design a user interface, considering the visual, cognitive, and physical considerations of humans and environmental and social issues.

The course is presented fully online through the university LMS (Learning Management System). Lessons were created for each chapter of the prescribed textbook; each lesson starts with a video providing the outcomes of the chapter followed by a presentation related to theoretical work. Once a certain amount of work has been covered, the student must complete a quick check consisting of 5 "match the terms" questions to see if they have grasped certain concepts. Students cannot move on to the next section of the lesson unless they have successfully completed the quick check. The next section normally contains a video based on the practical example of coding in the prescribed book. The students are shown what to do, how to do it, and where to do it while explanations are given verbally and in a speech bubble as to why they are doing it. Students are encouraged to complete the practical example of the code on their own once they have watched the video. The theory video, quick check, and practical video are repeated until the end of the chapter is reached. Students can download all the provided videos for offline viewing. Each presentation is also made available below the video as a PDF which can be downloaded. The complete code for each chapter's practical example as well as other self-assessment exercise solutions are made available for download as a ZIP file below the video of the final practical video. Links to Open Education Resources (OERs) regarding specific topics are also made available in the "Additional Reading section" of each Lesson. Considering the rapid changes in the World Wide Web (WWW) environment these links provide the most up-to-date information regarding a specific topic.

The primary objective of the assessment is to select a strategy that best reflects the course's outcomes. Additionally, the selection of assessment methods must be compatible with the institution's systems and requirements (Goosen and Van Heerden, 2016). The specific outcomes, assessment criteria of this course, the assessment method, and the assessment tool as available in the institution are listed in Table 1.

Table 1: Course Assessment

Specific outcomes	Assessment criteria	Assessment Method	Assessment Tool
Demonstrate an understanding of the concepts of Interface Design in view of its importance to the success of a computer-based product	• An understanding of what a user interface is and how it should be developed • The descriptions of the concepts of a graphical user interface were correct and according to the theory • The descriptions of the intelligent user interface were correct and according to the theory	• Theoretical – Bloom's understanding and memorizing	• Moodle Quiz
Demonstrate a clear understanding of problem statements as stated by computer users in various industries.	• Language proficiency is clearly demonstrated	• Theoretical – Bloom's evaluation and analysing	• Moodle forum blog
Identify and apply User Interface design phases	• The different phases of the user interface design process were demonstrated	• Theoretical – Bloom's understanding and memorizing	• Moodle Quiz
Design a user interface, considering visual, cognitive, and physical considerations of humans should be considered when designing a user interface	• An understanding of the visual, cognitive, and physical considerations is demonstrated through documentation	• Practical – Bloom's applying and creating	• CodeGrade
Design a user interface taking environmental and social issues into account when designing a user interface	• An understanding of the environmental and social issues is demonstrated through documentation	• Practical – Bloom's applying and creating	• CodeGrade
Design and evaluate graphical user interfaces	• The structure of a graphical user interface is demonstrated using window, dialog boxes and menus • The user interaction with a graphical user interface is demonstrated using industry standards, corporate standards, and the development of a solution specific and to the satisfaction of the client • The presentation of a graphical user interface will be demonstrated by how the data is shown on screens and windows.	• Practical – Bloom's applying and creating	• CodeGrade

1.1 Background

The drop-out and failure rates of programming courses have been contentious issues for many years. Mathematical proficiency, programming language, learning styles, teaching methods, assessment methods, and intrinsic and extrinsic motivation have all been examined by researchers as contributing factors to drop-out and failure rates. Nearly two decades ago Bergin and Reilly (2005) noted that it was well-known "in the Computer Science Education … community that students have difficulty with programming courses and this can result in high drop-out and failure rates". A decade later Vihavainen, Airaksinen, and Watson (2014) stated that despite decades of research, "no factor to date has been shown to influence programming performance across a range of different teaching contexts". Goosen and Van Heerden (2018) investigated the uptake of different assessment approaches in programming courses and concluded "Even if the best-planned, supported and implemented assessment strategies were in place, but they were not supported by the institutional systems and students, they will not be effective in addressing high drop-out and failure rates."

Prior to 2020 the course was presented in two semesters (February – June and July – November), students were required to submit four formative assessments, three quiz and a project-based assessment, and to write a venue based summative assessment. In 2020, when the Covid pandemic started, the course was still presented in two semesters with the same formative assessments, the summative assessment was, however, conducted as an online written examination. The course underwent some extensive changes during 2021, it became a year course (March – November), continuous assessment was introduced, and students were required to submit six formative assessments, three quiz and three blog assessments and one summative project-based assessment. Auto-graded coding assessments was incorporated in the course in 2022 and students were required to submit nine formative assessments, three quiz, three blog and three auto-graded coding assessments as well as one manually graded summative project-based assessment.

The 2022 quiz assessments had 30 questions consisting of a combination multiple-choice, true or false and fill in the missing word covering 3 chapters of the prescribed book, thus 10 questions per chapter. The blog assessments required student to write a 300-word reflection on each chapter they studied, each assessment covered 3 chapters. The students also had to comment on a minimum of three of their fellow students' blogs. The Automatic Grading System (AGS) coding assessments were based on In the Lab projects in the prescribed textbook that provide step-by-step coding instructions, increases in difficulty, and requires students to create webpages based on what they learned in each chapter. For the summative assessment students had to contact a car repair service company in their area and obtain their permission to use their information to develop a website for them.

1.2 Aim of the Study

The study reported on in this paper is aimed at investigating perceptions of continuous assessment by seeking to answer the following research question:

- Can a CODeL institution foster an enabling environment for valid and authentic continuous assessment in a programming course?

2. Literature Review

The Moodle LMS platform allows for online activities like quizzes, which assists in tracking student performance and learning. There is a wide range of quizzes that can be created with various item types and settings. López-Tocón (2021) states that "The use of Moodle quizzes, as online activities, favours the implementation of a different educational methodology…" and warns that "…not all quizzes can be used as assessment tools given that the item type and the quiz settings play an important role." Students find the Moodle quiz tool easy to use and that it helps them study theoretical aspects (Romero, García and Ceamanos, 2021).

Students learn authentic and meaningful ways of thinking and practicing through blogging, by participating in an assessment technique that required them to actively consider the course material (Christie and Morris, 2021). This is supported by Van Heerden and Van Der Merwe (2014) who stated "…the implementation of knowledge blogging in an ODL environment is particularly well suited to introductory programming courses when such blogging demands reflective activities and continued engagement with the course work."

Thompson *et al.* (2021) found that "When utilised correctly, an AGS can be of huge benefit to the development of a student's knowledge, providing additional scaffolding. Through effective feedback, effective use of test cases and all the other considerations, a student will have access to what is essentially a "digital tutor"." Barlow *et al* (2021) states that "Faculty were able to assess learning objectives at a granular level with the scaffolded programming assignments provided, and the auto-grader provides frequent and timely formative feedback. Students and teachers alike report a positive user experience, as do cost savings and ease of use". The user experience or user interface that the students have created using web frameworks or languages like HTML, CSS, and JavaScript is frequently the most important aspect of the grade. From the CodeGrade AGS "We cannot normally rely on unit testing or (command line) input and output tests for this reason. Instead, we will need to investigate and test ways to automatically interact with their website's UI."(Hillenius, 2021).

Since students actively participate, learn by doing, put their learning into practice, and address real or simulated issues, project-based learning is perfectly suited to evaluation for learning in programming courses (Munje, 2022). It is possible for students to demonstrate in concrete ways that they have understood the theory and can use it in the actual world through project-based assessment, which goes beyond merely evaluating their knowledge (Cifrian *et al.*, 2020). The implementation of project-based learning and evaluation has been shown through study to increase the performance of students taking programming courses (Cifrian *et al.*, 2020).

There are three major consideration when assessing students, validity, authenticity and dishonesty (Gikandi, Morrow and Davis, 2011). They categorized validity of assessment in an online environment as (1) authenticity of assessment activities, (2) effective formative feedback, (3) multidimensional perspectives, and (4) learner support. The authenticity of online assessment is categorized as (1) opportunities for documenting and monitoring evidence of learning, (2) multiple sources of evidence of learning and (3) explicit clarity of learning goals and shared meaning of rubrics. Gikandi, Morrow, and Davis (2011) also indicated that dishonesty "…is closely related to the issues of validity and authenticity. This implies that within the context of online formative assessment, aspects of dishonesty can be addressed by enhancing validity and authenticity."

Defining continuous assessment is somewhat problematic as there is a large variety of implementation instances. The major differences in defining continuous assessment were "key purposes, methods employed, sources of the tools or items, frequency, and context" (Muskin, Joshua A and Ibe, 2017). For the purposes of this paper the definition provided by United Nations Educational, Scientific and Cultural Organization (UNESCO) will be used. Their In-Progress Reflection No. 13 on Current and Critical Issues in Curriculum, Learning and Assessment defines continuous assessment as follows: "identify the level of learning of individual students (and sometimes of a class or some other grouping) on different aspects of the curriculum. These occur either for summative or formative purposes, and often both." (Muskin, Joshua A and Ibe, 2017).

3. Theoretical framework

Constructivism is an educational theory that may be traced back to the work of Jean Piaget (Bakar, Mukhtar and Khalid, 2019). Constructivism is embedded in Vygotsky's theory (Kozulin *et al.*, 2003). The constructivist theory holds that learners are active participants in their learning journey and that knowledge is built via experiences. In this study, students gain knowledge by actively participating in continuous assessment tasks.

One of the concepts of continuous assessment is to improve or reinforce teaching and learning which may be accomplished by having students participate in relevant activities within the teaching and learning process. According to constructivists effective teaching and learning is associated with authentic assessment, learner-centred learning, and the facilitator being a guide in a collaborative learning process (Ahmad, Sultana and Jamil, 2020). Throughout this process, facilitators must provide students with immediate feedback so that students may examine the existing gap between current knowledge and intended learning goals (Riese and Bälter, 2022). Through the feedback process, continuous assessment functions as a guide to self-evaluation for both the facilitator and students.

4. Methodology

4.1 Research Design

A mixed method approach is adopted (Mertens, 2019). This study conducted a survey that enabled the collection of both qualitative and quantitative data. These methods were selected with the purpose to answer the identified research question.

4.2 The Population

The class consisted of 538 students who were invited to participate in the course review hosted as an anonymous questionnaire in Google Drive. There were 184 respondents who participated.

4.3 The Participants

The survey respondents were drawn from students of a first-year programming course at a CODeL university in South Africa. The participants were 184 first-year programming students in the academic year 2022, in the computer science department.

4.4 Data Collection Process

The study utilised qualitative and quantitative approaches. A survey was used to collect both qualitative and quantitative data that was based on open-ended (qualitative) and closed-ended (quantitative) questions (Mertens, 2019).

4.5 Instrument Design

The survey was designed according to the objective of the study and prepared after critically studying the literature review. The respondents were asked questions related to their perceptions regarding continuous assessment in a programming course. The survey form utilised open-ended questions (qualitative data) and closed-ended Likert scale questions (quantitative data). The scoring for the Likert scale was e.g., Fully disagree, Disagree, Neutral, Agree, and Fully agree.

5. Results

The average submission rate of the quiz assessment was 98% of the participants in the study. Most of the students disagreed with the statement that the quiz tool made it difficult to complete the assessment, 34% fully disagreed and 40% disagreed. The statement “It is easy to pass quiz assessments” elicited a 28% fully agree, 23% agree and 30% neutral response.

On average 85% of the students who participated in the study submitted their blog assessments. Participants disagreed with the statement “I cannot see how blogging about programming will assist my communication skills in the work environment.”, 25% fully disagreed, 29% disagreed and 23% was neutral. The participants agreed that the blog assessment assesses their insight into the work they studied, 20% fully agreed, 33% agreed and 24% was neutral.

The AGS coding assessments have the lowest submission rate at only 73% of participants having submitted the assessments. The assessment tool did not contribute toward the low submission rate as 23% of the participants totally disagreed with the statement that “CodeGrade made it difficult to submit the assessment”, 31% disagreed and 24% was neutral. The participants felt that CodeGrade assisted them in finding errors in the code with 18% fully agreeing, 27% agreeing and 24% being neutral.

The project-based assessment was submitted by 91% of the participants. The CodeGrade AGS was used for the submission of the assessments, however, the assessment was graded manually. The above answers as pertaining to CodeGrade are also relevant for this assessment.

The participants preferred continuous assessment over writing examinations as 45% fully agreed with the statement “I prefer continuous assessment to writing an examination for the course”, 17% agreed and 17% were neutral. The participant responses to the statement “I prefer viewer assignments and writing an examination” supports the first statements outcome as 35% fully disagreed with the statement, 24% disagreed and 24% were neutral. The participants agreed with the statement “The instructions on how to complete the assessments was clear”, with 32% fully agreeing, 37% agreeing and 17% was neutral. The statement “I feel the assessments for the course was setup in such a way that I can perform well if I apply myself” was responded to positively with 37% respondents fully agreeing, 32% agreeing and 16% being neutral.

6. Discussion

6.1 Validity and Authenticity of Quiz Assessment

Addressing the validity, authenticity and dishonesty in the quiz assessments, question pools was created for each chapter consisting of between 90 – 120 questions, ten questions per chapter were selected randomly for each student and multiple-choice answers were also randomized. Students received immediate feedback on their quiz assessments that provides explanation as to why a questions answer is considered wrong. Students were given three opportunities to complete the assessment and a new set of questions were generated for each opportunity. Through the randomization of both questions and answers the aspect of dishonesty is addressed.

6.2 Validity and Authenticity of Blog Assessment

Students were required to write a blog reflecting on what they have learned for the assessment based on the material they studied in specific chapters. Each reflection must be at least 600 words long, with 200 words for each chapter. For every three chapters that students have studied, they were required to submit one blog. In addition, students were required to leave comments on at least three other students' blogs. This gives them a chance to interact with other students and get a different perspective on the material. These assessments have a high validity and authenticity, which is crucial in the CODeL environment because no two students can have identical reflections on what they have learned.

6.3 Validity and Authenticity of Automated Grading System Assessment

CodeGrade was implemented as the AGS to grade the coding assessments. Students were given a specific set of instructions to follow to build a website, they were provided with image, sound, video, links, and detailed descriptions of how to write the code. After every third chapter in the prescribed book the students must submit the completed code for the three chapters. In CodeGrade web user interfaces were auto graded with Selenium and Jest. The specific requirements were used as a rubric against which the code is tested, and feedback provided. Feedback entails showing students the required code versus what they submitted, which allows them to make corrections to their code. Students were allowed three submissions and the grade of the final

submission is captured. As all students were required to submit the exact same code the assessment have a low validity and authenticity with a high possibility of dishonesty.

6.4 Validity and Authenticity of Project Based Assessment

The summative assessment of the course was project-based, students were required to contact a car repair shop in their area and develop a website for them based on given criteria. Students had to obtain images, sound, videos, and all other relevant information about the shop to incorporate into the website. The students were given three opportunities to submit their assessment via the CodeGrade AGS and the last submission was graded manually using a rubric that was also provided to the students. Although students may design the website for the same shop, their code and website should never look identical, if this is the case there is a high probability of dishonesty extending high validity and authenticity to this assessment.

6.5 Validity and Authenticity of Continuous Assessment

The continuous assessment strategy followed in the course was to implement different assessment methods to assess different aspects of the curriculum. Students were given explicit instructions on what will be assessed through each assessment method and how each assessment method works. The feedback received on certain assessments allowed students to return to the work they have studied and make corrections where necessary and then resubmit their work.

7. Conclusion

Ensuring valid and authentic continuous assessment of programming courses at a CODeL institution requires careful planning, attention to detail, and the use of appropriate assessment methods and tools (Van Zyl and Le Roux, 2019). The first step is to clearly define the learning outcomes for the programming courses and the assessment criteria that will be used to evaluate students' performance (Wallace, Gloria and Munetsi, 2022). This should be done at the beginning of the course and communicated clearly to the students.

It is important to use a variety of assessment methods to evaluate students' learning, such as quizzes, coding assignments, blogs, and online tests. This can help ensure that different aspects of the learning outcomes are being assessed and can also reduce the risk of cheating (Pereira, Flores and Niklasson, 2016). Providing clear instructions for all assessments and giving students detailed feedback on their performance are also essential. This will help them understand what they need to do to improve and will also help maintain the validity and authenticity of the assessment.

Regular monitoring of student progress can help identify issues early on and allow for intervention if necessary (Apaloo, Kombat and Mohammed, 2022). This can be done using learning analytics and online discussion forums. Academic integrity is crucial to ensure the validity and authenticity of assessments. This can be achieved by using appropriate assessment methods and tools, clearly communicating academic expectations to students, and taking appropriate actions if academic misconduct is detected.

Overall, ensuring valid and authentic continuous assessment of programming courses at a CODeL institution requires a proactive approach and the use of a range of strategies and tools to promote student learning and maintain academic integrity. CODeL institutions can foster an enabling environment for valid and authentic continuous assessment in a programming course.

References

Ahmad, S., Sultana, N. and Jamil, S. (2020) 'Behaviorism vs Constructivism: A Paradigm Shift from Traditional to Alternative Assessment Techniques Sadia Jamil', *Journal of Applied Linguistics and Language Research*, 7(2), pp. 19–33.

Apaloo, M., Kombat, D. and Mohammed, Y. (2022) 'Continuous Assessment As a Good Predictor of Actual Performance', *Education, Sustainability & Society*, 5(1), pp. 21–23. doi: 10.26480/ess.01.2022.21.23.

Bakar, M. A., Mukhtar, M. and Khalid, F. (2019) 'The development of a visual output approach for programming via the application of cognitive load theory and constructivism', *International Journal of Advanced Computer Science and Applications*, 10(11), pp. 305–312. doi: 10.14569/IJACSA.2019.0101142.

Barlow, M. *et al.* (2021) 'MOCSIDE: An Open-Source and Scalable Online IDE and Auto-Grader for Introductory Programming Courses', *J. Comput. Sci. Coll.*, 37(5), pp. 11–20.

Bergin, S. and Reilly, R. (2005) 'The influence of motivation and comfort-level on learning to program.', in *17th Annual Workshop of the Psychology of Programming Interest Group*. Sussex University: Psychology of Programming Interest Group, pp. 293–304.

Christie, H. and Morris, N. (2021) 'Using assessed blogs to enhance student engagement', *Teaching in Higher Education*. Taylor & Francis, 26(4), pp. 573–585. doi: 10.1080/13562517.2019.1662390.

Cifrian, E. *et al.* (2020) 'Integration of different assessment approaches: application to a project-based learning engineering course', *Education for Chemical Engineers*, 31, pp. 62–75. doi: 10.1016/j.ece.2020.04.006.

Gikandi, J. W., Morrow, D. and Davis, N. E. (2011) 'Online formative assessment in higher education: A review of the literature', *Computers & Education*. Elsevier Ltd, 57(4), pp. 2333–2351. doi: 10.1016/j.compedu.2011.06.004.

Goosen, L. and Van Heerden, D. (2016) 'E-learning environment tools to address online and open distance education context challenges', *Proceedings of the International Conference on e-Learning, ICEL*, 2016-Janua, pp. 275–284.

Goosen, L. and Van Heerden, D. (2018) 'Assessment of Students in Higher Education— Information and Communication Technology Tools and Tips', *Progressio*, 40(1 & 2), pp. 1–23. doi: 10.25159/0256-8853/4704.

Hillenius, D. (2021) *Autograding Web UI*. Available at: https://www.codegrade.com/blog/autograding-web-user-interfaces-with-selenium-and-jest-in-codegrade.

Kozulin, A. *et al.* (2003) *Vygotsky's Educational Theory in Cultural Context*. Cambridge University Press.

López-Tocón, I. (2021) 'Moodle quizzes as a continuous assessment in higher education: An exploratory approach in physical chemistry', *Education Sciences*, 11(9). doi: 10.3390/educsci11090500.

Mertens, D. M. (2019) *Research and Evaluation in Education and Psychology: Integrating Diversity ... - Donna M. Mertens - Google Books*. 5th edn. SAGE Publications. Available at: https://books.google.co.za/books?hl=en&lr=&id=6RR7DwAAQBAJ&oi=fnd&pg=PT17&dq=mixed+method+research+approach+computer+science+education&ots=dfl9FvB2fO&sig=e6ARz4sUZpGs2bmhJs6zyjLA2XU&redir_esc=y#v=onepage&q=mixed method research approach computer science e (Accessed: 14 March 2023).

Munje, R. K. (2022) 'Project-Based Learning: Teaching Methodology to Impart Knowledge and Skills', *Journal of Engineering Education Transformations*, 35(Special Issue 1), pp. 326–333. doi: 10.16920/jeet/2022/v35is1/22047.

Muskin, Joshua A and Ibe (2017) *Continuous assessment for improved teaching and learning: a critical review to inform policy and practice; Current and critical issues in curriculum, learning and assessment; Vol.:13; 2017*. 13. Available at: https://unesdoc.unesco.org/ark:/48223/pf0000255511.

Pereira, D., Flores, M. A. and Niklasson, L. (2016) 'Assessment revisited: a review of research in Assessment and Evaluation in Higher Education', *Assessment and Evaluation in Higher Education*. Routledge, 41(7), pp. 1008–1032. doi: 10.1080/02602938.2015.1055233.

Riese, E. and Bälter, O. (2022) 'A Qualitative Study of Experienced Course Coordinators' Perspectives on Assessment in Introductory Programming Courses for Non-CS Majors', *ACM Transactions on Computing Education*, 22(4), pp. 1–29. doi: 10.1145/3517134.

Romero, E., García, L. and Ceamanos, J. (2021) 'Moodle and Socrative quizzes as formative aids on theory teaching in a chemical engineering subject', *Education for Chemical Engineers*, 36, pp. 54–64. doi: 10.1016/j.ece.2021.03.001.

Thompson, A. *et al.* (2021) 'Considerations when using an Automatic Grading System within Computer Science Modules', *International Conference on Higher Education Advances*, pp. 589–597. doi: 10.4995/HEAd21.2021.13045.

Van Heerden, D. and Van Der Merwe, M. (2014) 'Employing objective measures in search of a relationship between knowledge Blogs and introductory programming performance outcome', *Proceedings of the International Conference on e-Learning, ICEL*, pp. 185–189.

Van Zyl, C. J. and Le Roux, W. J. (2019) 'Article title: Implementing Continuous Assessment in an Academic Programme for More Effective Learning Implementing Continuous Assessment in an Academic Programme for More Effective Learning', *UnisaRxiv*, (May), pp. 0–2. doi: 10.25159/UnisaRxiv/000006.v1.

Vihavainen, A., Airaksinen, J. and Watson, C. (2014) 'A systematic review of approaches for teaching introductory programming and their influence on success', in *Proceedings of the tenth annual conference on International computing education research - ICER '14*, pp. 19–26. doi: 10.1145/2632320.2632349.

Wallace, M. T., Gloria, M. L. and Munetsi, H. W. (2022) 'Continuous Assessment Learning Activity : an Overdue Progressive Assessment Approach Compromisingly Implemented', *International Journal of All Research Writings*, 4(July), pp. 57–68.