A Call for Critical Technology to Enable Innovative and Alternative Grading Practices

Adrienne Decker
Engineering Education
University at Buffalo
Buffalo, NY, USA
adrienne@buffalo.edu

Stephen H. Edwards
Computer Science
Virginia Tech
Blacksburg, VA, USA
edwards@cs.vt.edu

Bob Edmison
Computer Science
Virginia Tech
Blacksburg, VA, USA
kedmison@vt.edu

Manuel A. Pérez-Quiñones
Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC, USA
perez.quinones@charlotte.edu

Audrey Rorrer
Computer Science
University of North Carolina at Charlotte
Charlotte, NC, USA
audrey.rorrer@charlotte.edu

ABSTRACT

The call for alternative grading practices has been made both inside and outside the computing education community. Various practices exist to provide assessment and feedback to students that do not rely strictly on points out of one hundred percent, weighted averages, high stakes assignments, and grading for behaviors instead of learning. However, modern classrooms, especially computer science classrooms, rely on a myriad of digital tools to organize and maintain the course structure. Tools like learning management systems, automatic grading systems, submission systems, and practice systems all exist for computing students and faculty to use to help support the learning of programming concepts. By and large, these systems all rely on an underlying mechanism of points and aggregating points for scoring. In the face of such technological choices, adopting alternative grading practices can prove challenging for instructors and confusing for students. In this position paper, we advocate addressing key research problems to make these systems easier to use with alternative grading practices. These include comprehensive support for categorical grading, comprehensive support for rework and resubmission, and improved protocols for communication of scores and feedback. We propose an extension to LTI to support the needs of alternative grading practices, and we provide an initial design for this LTI extension. We discuss current problems and potential solutions and challenge the community to work on these problems and consider the design of future systems to embrace grading approaches that go beyond just points-based scoring.

SIGCSE TS 2026, February 18-21, 2026, St. Louis, MO, USA.

ACM ISBN 979-8-4007-2256-1/2026/02
https://doi.org/10.1145/3770762.3772530

CCS CONCEPTS

•Social and professional topics~Professional top-ics~Computing education~Student assessment•Social and professional topics~Professional topics~Computing education~Computing education programs~Computer science education•Social and professional topics~Professional topics~Computing education~Computing education programs~Computational science and engineering education

KEYWORDS

Alternative grading, tool support for grading

ACM Reference format:
Adrienne Decker, Stephen H. Edwards, Bob Edmison, Manuel A. Pérez-Quiñones, & Audrey Rorrer. 2026. A Call for Critical Technology to Enable Innovative and Alternative Grading Practices. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2026), Feb. 18–21, 2026, St. Louis, MO, USA. ACM, NY, NY, USA, 7 pages.* https://doi.org/10.1145/3770762.3772530

1 Introduction

There has been a growing call for faculty in computing to adopt alternative grading practices [17, 55]. Alternative grading prac-tices have been shown to enhance student learning and provide a better classroom experience. Decker et al. (2024) argue for the wider adoption and use of such practices at all levels of computing classrooms and have begun work on creating a community of practice around adopting such practices [11].

However, lurking under the surface, there exist critical tech-nology barriers that may be hindering adoption by computing faculty, as well as faculty from other disciplines. Traditional points-based grading practices are so ingrained in current educational practice that teaching tools take this approach for granted.

Everything, including learning management systems, gradebooks, online practice tools, and automated grading tools, are built to facilitate the traditional approach. In many cases, tools do not even allow the option of exploring alternative grading approaches. Indeed, the most widely used interoperability protocol for learning tools (LTI, discussed in Section 4.3) is predicated on this assumption, forcing learning tools to communicate grades on each activity as single numeric percentage score. Worse still, this limitation of existing educational tools presents a significant barrier to any educator who wishes to use or experiment with more innovative grading practices. These limitations are built into a wide range of learning tools, including general purpose tools such as Piazza [52], Gradescope [19], and Top Hat [58], and textbook supplementary tools from publishers such as Pearson [48], Wiley [62], and McGraw-Hill [42], and STEM-oriented learning tools such as LabFlow [29], Labster [30], and WeBWorK [61].

Particularly in STEM fields with large enrollments such as computer science, educational tools are increasingly necessary to help educators cope with increasing class sizes and enable faster, more actionable feedback on student work. Because of this, solving the technological barriers to innovative grading practices is critical to the adoption of these practices by larger cohorts of faculty. To this end, we identify the three most significant needs for enabling progress in and adoption of alternative grading practices:

1. Comprehensive support for categorical grading at the level of individual learning objectives, the level of assignments, and the level of cumulative course grading.
2. Comprehensive support, beyond simple late policies, for the rework and resubmission of deliverables where the student wishes to demonstrate improved mastery.
3. Improved protocols for communication among educational tools (such as e-textbooks, LMSs, auto-graders, and more) that support tracking of individual learning objectives to support categorical grading schemes and objective-directed feedback for students.

These critical needs are described here with two goals in mind. First, identifying these technical obstacles to broader adoption of alternative grading practices serves as a call for computing education researchers to come up with practical solutions. Second, raising awareness among tool builders within our community will aid them in avoiding reproducing these issues in new tools they produce.

2 Grades and Grading Practices

In this section, we introduce some main tenets of alternative grading practices and present some arguments for adoption of these techniques. Numerous authors have pointed out the problems with traditional grading mechanisms including achievement distortion [46], extrinsic rather than intrinsic motivation [12, 17], the problems with percentage grading and grading scales [16, 17, 21, 22, 45, 46, 53, 54], and other practices labeled toxic [55]. We will not attempt to reargue their points here. Instead, we will present our premises for what grades should represent and how we can use some of these proposed techniques to achieve that.

If you ask several people what a particular grade, albeit course grade, assignment grade, or even score on a particular question, means you will likely get several answers. Grades are rankings. Grades represent knowledge. Grades rate a demonstration of skills. Grades represent adherence to behaviors. Grades represent a mixture of these things. When you ask folks how they arrive at final course grades, the answers are just as varied, but it is likely the case that course grades are a mixture of demonstrations of knowledge and skill along with rating of behaviors and sometimes attitudes. Often, they are also ranking, with rigid ideas of how many A's can be awarded and where cutoffs should be placed to ensure that not too many people are allowed into exclusive groups. Grades can gatekeep.

For the purposes of this paper and this argument, we are adopting the view that grades represent proficiency with skill. They do not reflect behaviors (enacted or not acted). They do not reflect rank. They do not reflect the speed at which someone learned a concept. There is no limit to how many skills one can demonstrate proficiency in. There is no limit to how many students can demonstrate proficiency at the highest level in that skill. With that framing in mind, we look at problems with current grading systems and practices.

2.1 Common Grading Practices

It is a common practice for faculty to assign penalties for late work, zeros for missed assignments, and average scores together at the end of the term to arrive at final grades. Assigning penalties for late work dilutes the value of the grade because the grade is not just assessing achievement; it is also assessing behavior. A student who submits a perfect assignment late is demonstrating proficiency, but for whatever reason, was not able to submit the assignment by the deadline. A penalty for that in the scoring creates a score that does not reflect proficiency in the skills demonstrated by the assignment, but an ability to behave in a certain way.

When you take such scores earned across the semester and average them, that overall grade reflects not just a proficiency of skills, but also a demonstration of various behaviors [46]. For computer science, this can include turning work in on time (and time management in general), correctly using the necessary submission tools, putting files in correct folders, including any required supplementary statements or output snapshots, and more—even though these may not relate to any specific learning goals. Further, averaging scores across a semester to compute final grades gives equal weight to early evidence of learning as it does for learning late in the term [55]. If the goal is learning (or proficiency of skill), then the speed at which it happens should not impact a student's grade.

We have also adopted the practice in many computer science classrooms of using arbitrary weighting schemes for different types of assignments. For example, assignments done outside of the classroom might not be done independently and thus be weighted lower than in-class work. Assignments done in the classroom (e.g., exams) may be viewed as more authentic. Thus, the final weight given to different categories of assignments (e.g., lab, homework, programming, exams) is determined not because of any learning objective measure but because of faculty trust/mistrust of our own instruments accurately reflecting our learning outcomes.

In addition to averaging final grades, we lose information about learning when points are aggregated in an individual assignment. Whenever a student earns full or partial credit for a particular question or subsection, we can consider that information about how much students understand the associated learning objective. However, once we take all those points and aggregate to an assignment grade, we have lost information about which areas students are struggling to learn and therefore are unable to appropriately address knowledge acquisition deficits. We are then unable to provide appropriate educational support and scaffolds for students struggling with particular concepts. It is the case that at some point all our scores need to aggregate to a final grade in a course, but where and when this aggregation occurs has a significant impact on the information that both the faculty and the students can act upon to enhance their learning.

2.2 Alternative Grading Scales

Various authors advocate for a reduced grading scale [16, 21, 54]. Feldman [17] points out that a 0-100 scale, with 0-59 being failure, tilts the scale towards failure. Furthermore, if multiple grades are averaged to gain a final score, a 0 in an assignment has an overweighted punitive effect on the student's score. Nilson [45] suggests a pass/fail (Satisfactory/Unsatisfactory) grading scale with opportunities for students to resubmit work with a failing grade, thereby providing additional opportunities for practice. Rapaport's triage grading [53] recommends a three-valued scale: full credit if substantially correct; minimal credit if substantially incorrect; emerging credit if the item is neither of the above.

Regardless of the exact implementation, each of these methods advocates for re-focusing the score a student receives into larger buckets that better correspond to their level of understanding of a topic, as opposed to trying to work the system for a few more points on an assignment graded on a 0-100 scale. That is, as opposed to students playing the partial credit game, whereby their goal is to maximize points without changing effort (e.g., bargaining to get more points in an assignment) [45], smaller grading scales re-focuses student efforts on demonstration of knowledge to earn the score.

2.3 Resubmission to Demonstrate New Proficiency

The opportunity for students to resubmit work that does not sufficiently demonstrate comprehension or understanding of a concept within a grading system is key to furthering an emphasis on learning. Resubmission of an assignment lowers the stakes of the assignment, provides students with additional opportunities to practice and complete the assignment, and more closely matches the learning outcomes of the assignment or course [45]. The approach of pass/fail and multiple submissions encourages high standards and low stakes, an approach that Bowen [6] argues is best for student's learning.

2.4 CS Education

In the last few years, the CS Education community has begun to acknowledge some of the grading problems presented by Feldman [17] and others. In the SIGCSE TS 2020 panel "To Grade or Not To Grade" [64], the panelists acknowledged that they "love to teach" but "hate to grade." The reasons presented in the panel discussion resonate with some of the literature in education; "punitive measures" such as assigning a low grade to a student for behavioral reasons (e.g., misunderstanding the assignment) are not rewarding for teachers. Berns and East [2] discussed the idea of using equitable grading to design a computer science course. The ideas of equitable grading [17] are used to design both the assessment and the instruction for a given course. Success has been reported along several student metrics including improving student motivation, increasing assessment accuracy, and a decrease in concerns of plagiarism [2].

Specifications grading, which is a specific implementation of the mastery grading approach, is not completely new to the CS Education community. Posters [31] and several workshops [34, 35, 36, 37, 38] have been presented at the SIGCSE Technical Symposium on Computer Science Education on this topic for the last five years. There were also Birds of a Feather sessions devoted to alternative grading practices in the last three conference iterations [32, 33, 49]. In 2024, one of the Best Papers for the entire symposium was dedicated to the issues of alternative grading practices [11].

It is also important to note that the CC2020 curriculum overview report [1] extols the virtues of competency-based grading for computer science courses, which has many of the same qualities as described for specifications grading [45] and triage grading [53]. The community is interested in these topics and in supporting their students using these practices.

3 Educational Technology and Tools for Computing Classrooms

If faculty have decided to adopt alternative grading practices for their classroom, they must now implement that for their students. A key component of the modern higher education classroom is educational technology and integrated learning tools. This is where faculty often come up against problems with implementing various policies of reduced grading scales and resubmission of work. There are a number of tools in use that all share these limitations: general purpose tools such as Piazza [52], Gradescope [19], Top Hat [58], Perusall [50], Hypothesis [24], InQuizitive [26], and TurnItIn [59]; textbook publisher tools such as Pearson MyLab and Mastering [48], WileyPLUS [62], Cengage Learning [8], Macmillan Achieve [41], McGraw-Hill ALEKS [42], and Zybooks [65]; and STEM-oriented learning tools such as LabFlow [29], Labster [30], WeBWorK [61], and WirisQuizzes [63].

One of the most common types of technology in education is the learning management system (LMS), used in most modern classrooms in some way to manage student access to course content and grades. Canvas [7], Moodle [44], and Blackboard [3] are common examples. The adoption of LMSs by K-20 institutions has grown quickly since the first LMS was introduced 30 years ago [20]. The move to remote learning driven by the COVID-19 pandemic has only accelerated this growth [51]. Indeed, LMSs are now essentially ubiquitous, and considered essential tools for delivering courses in higher education [40]. LMSs are often the only "approved" way to disseminate grades to students, so the way that LMSs handle scoring has an immediate and direct impact on student learning.

An additional type of educational technology that has become very popular are electronic textbooks, e-textbooks, or e-books. Some instructors have developed their own [57], while others have adopted versions of popular paper textbooks that are now widely available from textbook publishers. These e-books offer both an authoring platform, as well as additional features including inline quizzes, interactive demonstrations of the book content, programming exercises, and LMS integration.

3.1 Educational Tools Specific to Computing Education

CS educators often use automated grading systems [18, 23, 24, 28] and small exercise practice systems in their classrooms, often to cope with soaring student enrollments [47]. Programming exercises, a common assignment in computing courses, are very amenable to using automation for assessment. Autograders can be tailored to specific languages, or can be more generalizable, handling programs in many different languages. Even with this flexibility, there is still a limitation when it comes to grades: only point-based grading is typically supported. Indeed, in a recent review of widely used autograding systems [4], none supported anything other than a points-based approach. There are both open-source solutions, such as Web-CAT [14, 60], and INGInious [25], and commercial solutions, including Gradescope [19] and CodeGrade [9], that are commonly used in classrooms.

Similarly, classrooms are starting to see increased usage of homework practice systems for small coding exercises. CodeWorkout [10, 15] is a stand-alone practice system, providing support for drill-and-practice questions in Java, Python, and C++. Several online textbooks marketed towards computer science classrooms combine features of the traditional e-book and these submission and practice systems. Zybooks [65] and Runestone [56] both provide the ability for students to practice the topics of a particular chapter by writing code within the online environment. Such systems can be used to provide syntax practice exercises [13, 39] as well as homework on basic programming skills. Such systems typically provide results using numeric scores on a 0-100 scale.

4 Supporting Alternative Grading Practices

Based upon our classroom experiences and those described in the computer science education literature, there is one resounding pain point for instructors seeking to adopt alternative grading practices, namely the lack of support for these practices in current classroom tools.

Here, we describe the three most significant areas where tool support can be improved: comprehensive support for categorical grading at all levels, comprehensive support for rework and resubmission of assignments or assessments, and improved protocols that support finer grained assessment and feedback.

4.1 Supporting Categorical Grading Schemes

To foster adoption of alternative grading practices, it is essential for learning management systems (LMSs) to fully support categorical grading scales, not only for each assignment but also at the level of individual learning objectives. Current LMSs fall short. While some LMSs allow some forms of specifications grading, their implementation is often limited. They might support specifications grading for specific assignment types but not others and often fail to support tracking learning objectives or specifications across individual questions or tasks within a single assignment, quiz, or exam.

LMSs also act as the central hub of an ecosystem of third-party learning tools that support a wide variety of learning activities. However, there is no support for categorical grading or for tracking individual learning objectives through such external tools, either. When assessments are broken down by learning objectives, instructors (and learning tools) can provide more targeted feedback. This helps students understand their strengths and areas for improvement on a more granular level, rather than seeing a single score that may not reflect specific skills or knowledge areas.

Similarly, meaningfully aggregating measures of individual learning objectives across multiple assignments is the cornerstone of mastery-based approaches. Such strategies are necessary to provide a comprehensive view of student progress, to encourage deeper learning, and to offer targeted feedback that highlights specific strengths and areas for improvement. This shift away from aggregating points will help educators and students focus on meaningful learning, ultimately fostering a more effective educational experience. By focusing on the mastery of individual learning objectives, educators can better support student growth and development, ensuring that students truly understand and can apply what they have learned.

To truly support alternative grading practices, LMSs must offer comprehensive support for categorical grading scales and adopt new methods for combining these grades into overall course evaluations.

4.2 Supporting Rework and Resubmission

Alternative grading practices frequently include support for reworking, retrying, and resubmitting assignments—key features for mastery-based learning. These options let students learn from mistakes, deepen understanding, and show improved mastery. Resubmissions also enable more personalized feedback, build resilience, and encourage a growth mindset. In contrast, most LMSs and learning tools offer limited support for resubmitting graded work, usually just allowing multiple submissions or deadline extensions. For quizzes or exams, students may receive grades and feedback before additional attempts, but for other assignments, the LMS simply collects submissions, with grading and feedback depending on human graders.

Online autograders and practice systems provide rapid feedback and allow students to make multiple submissions before an assignment deadline so that students can improve. However, feedback is often a numeric score or a report of errors, focused on reporting of results rather than mastery of learning objectives. To fully embrace alternative grading practices, LMSs and learning tools must evolve to support these iterative processes, ensuring a more effective and educational experience where students can continually improve.

4.3 Improving Protocol Support

As introduced in Section 2, a key limitation within the LMS and learning tool ecosystem is the Learning Tools Interoperability (LTI) protocol typically used to communicate between learning tools and an LMS. The LTI specification reduces scores on an assignment or activity to a single numeric percentage. This significantly contributes to the lack of support for alternative grading schemes in learning tools that interact with LMSs through this protocol. It is crucial to address this mismatch to provide better support for using such learning tools with alternative grading practices.

Also discussed in Section 4.1, comprehensive support for categorical grading scales at the level of individual learning objectives within an assignment, rather than relying on one assignment-level score, is also critical. While learning tools offer differing levels of support for recording such information, the limitations of the LTI protocol cut off the communication of such information back to the LMS, posing serious challenges for how student performance at the level of individual learning objectives can be aggregated or communicated. While we focus on the LTI protocol in this paper, other common EdTech protocols also demonstrate similar limitations. Addressing these shortcomings is a critical need.

5 Bridging the Gap from Current Tools: A Path Forward

Addressing the technological challenges described in Section 4 so that solutions are available is critical to enable greater adoption of alternative grading strategies. Without such solutions, the educational technology we rely on serves as a potent obstacle. Therefore, we need to create technological solutions for learning management systems and the external learning tools that work with them to support alternative grading practices, including automated graders, online practice systems, and electronic textbooks.

5.1 Mapping Learning Tool Results onto Alternative Grading Practices

As described earlier, learning management systems (LMS) have become the central hub for instructors to disseminate course materials, and for students to interact with those resources. As such, LMS developers have developed integration methods that can be used to extend the basic functionality provided by a vendor's tool. One of the most widely adopted standards for extended LMSs is the Learning Tools Interoperability (LTI) specification. This standard, developed by a consortium of industry and university organizations, specifies how an LMS and an LTI-compliant external system can securely exchange information, including class rosters and student grades.

However, a significant limitation of the LTI specification is that LTI-based grade exchanges happen using a points-based scheme. Specifically, scores in LTI grade exchanges are represented as floating point values between 0.0-1.0, although the most recent version of LTI allows additional comments or feedback to be passed alongside the scores. To support alternative grading practices across LTI-based learning tools, we must provide a clear way for instructors to describe how they wish to use this interface specification within the context of their own alternative grading approach.

For example, a simple approach to begin with is to have an instructor identify the number of levels in their grading scheme and then specify numeric subranges for each grading level. For example, an instructor could use a 4-valued scale like that proposed by Rapaport [53]. Raw points coming from an LTI tool can be divided into 4 separate score ranges to identify the corresponding level on the 4-valued scale, which can then be represented as 0, 1, 2, or 3 when being sent to the LMS. Additionally, the label corresponding to the earned level on the scale can be specified. The proxy would convert points to the fixed value based on the identified point ranges or cutoffs:

- 90 to 100 points (>= 0.9) map to a scaled score of 3, labeled “Full credit” or “Excellent”
- 70 to 89 points (>= 0.7 and < 0.9) map to a scaled score of 2, labeled “Emerging credit” or “Meets Expectations”
- 5 to 69 points (>= 0.05 and < 0.7) map to a scaled score of 1, labeled “Minimal credit” or “Needs Revision”
- 0 to 4 points (< 0.05) map to a scaled score of 0, labeled “Zero credit” or “Not Assessable”

This mapping could be represented as structured data in JSON, YAML, or another convenient format (here shown in YAML Flow format for brevity):

```
[{ level: "Not Assessable",  score_required: "0%" },
 { level: "Minimal Credit",  score_required: "5%" },
 { level: "Emerging Credit", score_required: "70%" },
 { level: "Full Credit",     score_required: "90%" }]
```

For tools that provide richer results beyond simply a numeric score, specific exercises, questions, or features that map to learning objectives identified by the instructor may be required by the instructor to meet certain levels. For example, an instructor may wish to require students to successfully answer questions 1-5 on a quiz to receive full credit, in addition to earning a minimum score overall:

```
[{ level: "Not Assessable",  score_required: "0%" },
 { level: "Minimal Credit",  score_required: "5%" },
 { level: "Emerging Credit", score_required: "70%" },
 { level: "Full Credit",     score_required: "90%",
   required: ["Q1", "Q2", "Q3", "Q4", "Q5"] }]
```

For tools that provide for naming or tagging of specific exercises or features, this approach can be expanded to allow the use of regular expression patterns or other mechanisms.

5.2 Handling Rework and Resubmission

Resubmission of work to show improved mastery is a core aspect of many alternative grading practices. While many educational tools (and LMSs) support a basic model of allowing late submissions with or without a grade penalty, few go beyond that. Educators using alternative grading practices instead have to employ their own manual strategies for how to allow students to resubmit work. Nilson [45], for example, suggests giving students several "resubmission tokens" that they can use to resubmit work that received a low grade, and other similar approaches have been described in the literature.

In working towards a solution, it is possible to develop a simple LTI-based tool that allows instructors to set up a predefined number of resubmission tokens for students, allows students to "cash in" a token to resubmit a particular assignment, and then uses a tool-specific API (such as the Canvas API) to add an individual extension for that student to resubmit work. Dashboards or automatic notifications for course staff when work has been resubmitted and needs regrading are also possible. By leveraging existing facilities within autograders or practice systems to allow multiple submissions and set individual student extensions, it is possible to extend that support to cover resubmission or work at a later date in a natural way so that manual processes are largely eliminated, facilitating smoother adoption.

5.3 Improving Protocol Support

While the LTI protocol has improved interoperability among educational tools, at its core it forces grades to be represented as single numeric scores. Fortunately, there are other protocols for interoperability, and LTI itself has numerous custom extensions that have been implemented by different LMSs. Such extensions allow external tools to provide feedback in richer forms alongside numeric scores, whether through supplementary textual content, or URLs used to present tool-generated feedback in a more sophisticated way.

In the future, educational researchers can work on new generations of protocols that address a wider variety of grading strategies. In the meantime, a proxy service that sits between and mediates communication between LTI tools and LMSs can provide a powerful capability for actively improving support for alternative grading practices. Such a proxy service would be a drop-in method for implementing the score mapping strategy described in Section 5.1 for handling reduced grading scales. Further, if instructor-provided score mapping schemes include specifications of individual learning objectives identified by the instructor, such a proxy service could provide feedback structured around learning objectives demonstrated or needing improvement, piggybacking on LMS-specific extensions to pass on this feedback where practical. Such a service is a useful intermediate goal while the community grapples with the longer-term evolution of the interoperability protocols on which our tools rely.

6 Conclusion

While many faculty want to adopt alternative grading practices in their classrooms, there are still several technological barriers to adoption that need to be addressed. Here, we have identified the three most pressing needs for enabling progress in and adoption of alternative grading practices: comprehensive support for categorical grading, comprehensive support for rework and resubmission, and improved protocols for communication of scores and feedback.

We need researchers to address these obstacles to alternative grading practices, and we also need members of our own community developing tools to be aware of the issues and how to avoid them. This includes clear strategies for supporting categorical grading at the level of individual learning objectives as well as for whole assignments, together with aggregation approaches that allow summarizing student performance on learning objectives as well as formulating meaningful course-level grades. We also need more comprehensive approaches to supporting the full range of resubmission policies that instructors advocate, going beyond simple late policies. To facilitate interoperability among the various learning tools in our classrooms, we need improved protocols that allow communication of more comprehensive assessment results at the level of individual learning objectives, rather than reducing all assignment grades to single numbers.

Feedback at a more micro and formative level while the students are engaged in the learning process but have not yet reached mastery can help them develop better awareness about what they know and don't yet know. Alongside the tool support, we need research into the types of feedback that are useful during this process, particularly when learning a skill like programming.

If we can do this, faculty can elevate their teaching, and we begin to build towards a place where students are motivated by learning and not about chasing the points to get all 100 of them.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. National Science Foundation under Grant Nos. 2235337, 2235643, 2235644, 2418493, and 2418494.

REFERENCES

[1] ACM and IEEE-CS. Computing Curricula 2020: Paradigms for Global Computing Education. Association for Computing Machinery CC2020 Task Force. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf

[2] Berns, A., and East. J.P., 2020. Toward a Pedagogy Compatible with Equitable Grading: Tenets, Difficulties, and Suggestions. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 1358. https://doi.org/10.1145/3328778.3372675

[3] Blackboard Learn. https://www.anthology.com/products/teaching-and-learning/learning-effectiveness/blackboard-learn, last accessed Jan 15, 2025.

[4] Blanchard, J., Hott, J.R., Berry, V., Carroll, R., Edmison, B., Glassey, R., Karnalim, O., Plancher, B., and Russell, S., 2022. Leveraging Community Software in CS Education to Avoid Reinventing the Wheel. In Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 2 (ITiCSE '22). Association for Computing Machinery, New York, NY, USA, 580–581. https://doi.org/10.1145/3502717.3532169

[5] Bloom, B. S. 1968. "Learning for Mastery". Evaluation Comment 1, 2 (March 1968), 12 pages.

[6] Bowen, J.A., 2012. Teaching naked: How moving technology out of your college classroom will improve student learning. John Wiley & Sons.
[7] Canvas for Higher Education. https://www.instructure.com/higher-education/products/canvas last accessed January 15, 2025.
[8] Cengage Learning. https://www.cengage.com/, last accessed on January 15, 2025
[9] CodeGrade. https://www.codegrade.com/, last accessed January 15, 2025.
[10] CodeWorkout. https://codeworkout.cs.vt.edu/, last accessed January 15, 2025.
[11] Decker, A., Edwards, S. H., McSkimming, B. M., Edmison, B., Rorrer, A., and Pérez-Quiñones, M. A. 2024. Transforming Grading Practices in the Computing Education Community. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE 2024). ACM, New York, NY, USA, 276 - 282. https://doi.org/10.1145/3626252.3630953
[12] Dweck, C.S., 2007. Mindset: The new psychology of success. Random House, New York, NY, USA.
[13] Edwards, J., Ditton, J., Trninic, D., Swanson, H., Sullivan, S., and Mano, C., 2020. Syntax exercises in CS1. In Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER '20) ACM, Virtual Event, New Zealand, 216-226. http://dx.doi.org/10.1145/3372782.3406259.
[14] Edwards, S.H., 2004. Using software testing to move students from trial-and-error to reflection-in-action. In Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04) ACM, Norfolk, Virginia, USA, 26-30. DOI= http://dx.doi.org/10.1145/971300.971312.
[15] Edwards, S.H. and Murali, K.P., 2017. CodeWorkout: Short Programming Exercises with Built-in Data Collection. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17) ACM, Bologna, Italy, 188-193. DOI= http://dx.doi.org/10.1145/3059009.3059055.
[16] Erickson, J. A., 2010. Grading Practices: The Third Rail. Principal Leadership, v10 n7 p22-24, 26 Mar 2010
[17] Feldman, J., 2019. Grading for equity: What it is, why it matters, and how it can transform schools and classrooms. Corwin Press, Thousand Oaks, CA, USA.
[18] Forsythe, G.E., and Wirth, N. 1965. Automatic Grading Programs. Commun. ACM 8, 5 (may 1965), 275–278. https://doi.org/10.1145/364914.364937
[19] Gradescope. https://www.gradescope.com/, last accessed January 15, 2025.
[20] Gumasing, M.J.J., Vasquez, A.B., Doctora, A.L.S., and Perez, W.D.D., 2022. Usability Evaluation of Online Learning Management System: Comparison between Blackboard and Canvas. In 2022 The 9th International Conference on Industrial Engineering and Applications (Europe) (Barcelona, Spain) (ICIEA-2022-Europe). Association for Computing Machinery, New York, NY, USA, 25–31. https://doi.org/10.1145/3523132.3523137
[21] Guskey, T.R., 2013, "The Case Against Percentage Grades" (2013). Educational, School, and Counseling Psychology Faculty Publications. 22. https://uknowledge.uky.edu/edp_facpub/22
[22] Guskey, T.R., 2000. Grading policies that work against standards ... and how to fix them. NASSP Bulletin, 84(620), 20–29.
[23] Hollingsworth, J., 1960. Automatic Graders for Programming Classes. Commun. ACM 3, 10 (oct 1960), 528–529. https://doi.org/10.1145/367415.367422
[24] Hypothesis. https://web.hypothes.is/, last accessed on January 15, 2025
[25] INGInious. https://github.com/UCL-INGI/INGInious, last accessed Jan 15, 2025.
[26] InQuizitive. https://wwnorton.knowledgeowl.com/help/inquizitive, last accessed on January 15, 2025
[27] Instructure Community. How do I use the Learning Mastery Gradebook to view outcome results in a course? https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-use-the-Learning-Mastery-Gradebook-to-view-outcome/ta-p/775, last accessed January 15, 2025.
[28] Isong, J., 2001. Developing an Automated Program Checkers. Journal of Computing Sciences in Colleges 16, 3 (2001), 218–224.
[29] LabFlow. https://labflow.freshdesk.com/, last accessed on January 15, 2025
[30] Labster. https://www.labster.com/, last accessed on January 15, 2025
[31] Largent, D.L., 2022. Instantiating Specifications Grading in Computer Science Courses. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2022). Association for Computing Machinery, New York, NY, USA, 1118. https://doi.org/10.1145/3478432.3499132
[32] Largent, D., Decker, A., Edwards, S., Pérez-Quiñones, M.A., and Roberson, C. 2025. BOF: Grading for Equity in Computer Science Courses. In Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2025). Assoc for Computing Machinery, New York, NY, USA, (in press).
[33] Largent, D., Pérez-Quiñones, M.A., Roberson, C., Wilson, L., Decker, A., and Edwards, S. 2024. BOF: Grading for Equity in Computer Science Courses. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 1241. https://doi.org/10.1145/3545947.3573330.
[34] Largent, D.L, and Roberson, C., 2020. Transform Your Computer Science Course with Specifications Grading. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 1396. https://doi.org/10.1145/3328778.3367002
[35] Largent, D.L., Roberson, C., and Sgro, C., 2021. Transform Your Computer Science Course with Specifications Grading. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21). Association for Computing Machinery, New York, NY, USA, 1355. https://doi.org/10.1145/3408877.3432505
[36] Largent, D.L., Roberson, C., and Sgro, C., Pérez-Quiñones, M., and Wilson, L.F., 2022. Transform Your Computer Science Course with Specifications Grading. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2022). Association for Computing Machinery, New York, NY, USA, 1200. https://doi.org/10.1145/3478432.3499167
[37] Largent, D.L., Roberson, C., Wilson, L., and Pérez-Quiñones, M. 2024. Transform Your Computer Science Course with Specifications Grading. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA , 1200. https://doi.org/10.1145/3478432.3499167
[38] Largent, D.L., Roberson, C., Wilson, L., Sgro, C., Pérez-Quiñones, M., and Wilson, L. 2023. Transform Your Computer Science Course with Specifications Grading. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 1186. https://doi.org/10.1145/3545947.3569614.
[39] Leinonen, A., Nygren, H., Pirttinen, N., Hellas, A., Leinonen, J., 2019. Exploring the Applicability of Simple Syntax Writing Practice for Learning Programming. In Proceedings of the Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19) (Minneapolis, MN, USA2019), ACM, 84–90. DOI= http://dx.doi.org/10.1145/3287324.3287378.
[40] Liaw, S., 2008. Investigating students' perceived satisfaction, behavioral intention, and effectiveness of e-learning: A case study of the Blackboard system. Computers & Education 51, 2 (2008), 864–873. https://doi.org/10.1016/j.compedu. 2007.09.005
[41] Macmillan Achieve. https://www.macmillanlearning.com/college/us/digital/achieve, last accessed on January 15, 2025
[42] McGraw-Hill ALEKS. https://www.aleks.com/, last accessed on January 15, 2025
[43] Moodle. Competencies - MoodleDocs. https://docs.moodle.org/310/en/Competencies, last accessed on Jan 15, 2025.
[44] Moodle - Open-source learning platform. https://moodle.org/, last accessed on January 15, 2025.
[45] Nilson, L.B., 2015. Specifications grading: Restoring rigor, motivating students, and saving faculty time. Stylus Publishing, LLC, Sterling, VA, USA.
[46] O'Connor, K., 2012. Fifteen Fixes for Broken Grades: A Repair Kit. Toronto ON: Pearson Canada; 2012.
[47] Paiva, J.C., Leal, J.P., and Figueira, A., 2022. Automated Assessment in Computer Science Education: A State-of-the-Art Review. ACM Trans. Comput. Educ. 22, 3, Article 34 (jun 2022), 40 pages. https://doi.org/10. 1145/3513140
[48] Pearson MyLab and Mastering. https://mlm.pearson.com/, last accessed on January 15, 2025
[49] Pérez-Quiñones, M., Largent, D.L., Moosvi, F., Roberson, C., Sgro, C., Toti, G., and Wilson, L. F.. 2023. BOF: Grading for Equity in Computer Science Courses. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 1241. https://doi.org/10.1145/3545947.3573330
[50] Perusall. https://www.perusall.com/, last accessed on January 15, 2025
[51] Peters, D. 2021. Learning management systems are more important than ever. Retrieved 2022-07-08 from https://www.universityaffairs.ca/features/feature-article/learning- management- systems- are- more- important- than- ever/
[52] Piazza. https://piazza.com/, last accessed on January 15, 2025
[53] Rapaport, W.J., 2011. A triage theory of grading: The good, the bad, and the middling. Teaching Philosophy 34, 4, 347-372. DOI= http://dx.doi.org/10.5840/teachphil201134447.
[54] Reeves, D.B., 2004. "The Case Against the Zero," Phi Delta Kappan, Vol. 86, No. 4, December 2004, pp. 324-325.
[55] Reeves, D.B., 2008. Leading to Change: Effective Grading Practices. Educational Leadership, 65(5), 85-87.
[56] Runestone Academy. https://runestone.academy/, last accessed January 15, 2025.
[57] Shaffer, C.A., 2016. OpenDSA: An Interactive ETextbook for Computer Science Courses. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (Memphis, Tennessee, USA) (SIGCSE '16). Association for Computing Machinery, New York, NY, USA, 5. https://doi.org/10.1145/2839509.2850505
[58] Top Hat. https://tophat.com/, last accessed on January 15, 2025
[59] TurnItIn. https://www.turnitin.com/, last accessed on January 15, 2025
[60] Web-CAT: the Web-based Center for Automated Testing. https://web-cat.org/, last accessed on January 15, 2025.
[61] WeBWorK. https://openwebwork.org/, last accessed on January 15, 2025
[62] WileyPLUS. https://www.wileyplus.com/, last accessed on January 15, 2025
[63] WirisQuizzes. https://www.wiris.com/en/wirisquizzes/ last accessed Jan 15,2025
[64] Wolz, U., Carmichael, G., Garcia, D., MacKellar, D., and Veilleux, N., 2020. To Grade or Not To Grade. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 479–480. https://doi.org/10.1145/3328778.3366978
[65] Zybooks. https://www.zybooks.com/, last accessed on January 15, 2025