

RESEARCH ARTICLE

Validity of Log-Based Indicators of Algorithmic Competence in An Adaptive Programming Environment: A Simulation Study

 **Nazokat R. Kayumova**

Jizzakh Branch of the National University of Uzbekistan named after Mirzo Ulugbek, Jizzakh, Uzbekistan

VOLUME: Vol.06 Issue03 2026

PAGE: 291-298

Copyright © 2026 European International Journal of Pedagogics, this is an open-access article distributed under the terms of the Creative Commons Attribution-Noncommercial-Share Alike 4.0 International License. Licensed under Creative Commons License a Creative Commons Attribution 4.0 International License.

Abstract

Background. Adaptive programming environments increasingly report students' competence from log data such as attempts, hints, solution time and written reflections. Such "digital trace" indicators are attractive for formative assessment, but their validity is rarely examined before they are shown to teachers and students. **Purpose.** This study examines whether the four indicators of a process-based learner model, implemented in the AlgoAdapt system to estimate the cognitive, practical, metacognitive and reflective components of algorithmic competence, measure the constructs they are intended to measure. **Methods.** Following an evidence-centred perspective on assessment, the scoring code of the system was applied without modification to logs generated for 3,000 synthetic learners whose four latent component traits were known. Convergent and discriminant validity, the influence of a construct-irrelevant trait (verbosity), parallel-forms reliability for windows of 10–80 attempts and robustness to two response strategies (keyword padding of reflections and copying generated solutions) were analysed. **Results.** All indicators correlated substantially with their target traits ($r = .78-.87$) and were reliable for the default window of 40 attempts ($r = .91-.96$). However, the cognitive and practical indicators were practically identical ($r > .99$), the metacognitive indicator was partly driven by practical skill ($\beta = .38$), and verbosity explained part of the reflective indicator ($\beta = .48$). Copying generated solutions moved all copiers into the "high" band on three components. **Conclusions.** Reliable process indicators are not automatically valid. We propose concrete revisions—separating evidence sources, adjusting time for correctness, rubric-based reflection scoring and process checks that resist copying—that should be made before such indicators inform grades.

KEY WORDS

Learning analytics; process data; evidence-centred assessment; validity; algorithmic competence; introductory programming; generative AI; simulation.

1. INTRODUCTION

Competence in algorithm design is a central outcome of introductory programming, yet it is difficult to assess with final answers alone. A correct program can hide a fragile understanding, and an incorrect one can conceal sound reasoning disrupted by a minor error (Lister et al., 2004; Robins et al., 2003). Automated assessment systems have therefore begun to record not only whether a solution passes

tests but also how it was reached: the number of submissions, the time taken, the hints requested and the errors made (Ihantola et al., 2010; Ihantola et al., 2015). Learning analytics promises to turn these traces into timely information for teachers and students (Siemens & Long, 2011), and adaptive systems use them to estimate competence and to choose the next task.

The assessment literature warns that this promise depends on validity. In the evidence-centred view, every assessment rests on a chain linking a model of the competence, the observations that are collected and the interpretation that turns observations into claims (Mislevy et al., 2003; Pellegrino et al., 2001). Validity is the degree to which evidence and theory support these interpretations for their intended use (AERA et al., 2014; Messick, 1995). Process indicators raise specific threats to this chain. They can confound the target competence with other characteristics (construct-irrelevant variance), they can fail to distinguish components that are supposed to be different, and, because students learn what the system rewards, they can be gamed (Baker et al., 2004). The arrival of generative AI tools adds a new threat: a student who pastes a generated solution produces a trace that may look like expert performance (Denny et al., 2024; Prather et al., 2023).

Empirical validation of process indicators with real students is necessary but slow, and it cannot reveal problems whose source is the scoring rule itself. Simulation offers a complementary first check: when synthetic learners with known latent traits produce logs, the scoring rules can be examined against the “ground truth” that real data never provide. Such simulation studies are standard in psychometrics for evaluating estimators before field use (Feinberg & Rubright, 2016; Morris et al., 2019).

This study applies such a check to the learner model of AlgoAdapt, an adaptive system for first-year programming courses that estimates four components of algorithmic competence—cognitive, practical, metacognitive and reflective—from the attempt log and shows them to teachers on a dashboard. Four research questions guided the study:

RQ1. Do the four indicators correlate with the latent traits they are intended to measure (convergent validity), and do they distinguish between components (discriminant validity)?

RQ2. To what extent are the indicators influenced by a construct-irrelevant characteristic, namely the tendency to write long reflections?

RQ3. How reliable are the indicators, and how does reliability depend on the number of logged attempts used?

RQ4. How robust are the indicators to two response strategies: padding reflections with expected keywords and copying generated solutions?

2. METHODS

2.1. The learner model under study

AlgoAdapt 2.0 is a web-based adaptive system for the first-year courses “Programming 1–2”, organised into seven modules from linear algorithms to sorting and searching (registered as computer software, certificate No. DGU 64245). After each submission the system logs the result, the attempt number for the task, the solution time, the task level (which determines the normative time), the hint level and the student’s answer to a reflection prompt. For each student and module, the learner model computes four indicators from the 40 most recent attempts (Table 1); the integral indicator is their mean. Values of 0.75 and above are reported as “high”, 0.50–0.75 as “intermediate” and below 0.50 as “low”. The intended interpretation is that each indicator reflects the corresponding component of algorithmic competence (Table 1, last column).

Table 1. Indicators of the AlgoAdapt learner model and their intended interpretation

Indicator	Scoring rule (weights)	Intended construct
K_{cog}	$0.65 \times \text{share correct} + 0.35 \times \text{mean score}$	Knowledge of algorithmic concepts and structures
K_{prac}	$0.55 \times \text{mean score} + 0.25 \times \text{share correct} + 0.20 \times (1 - \text{normalised repeated attempts})$	Designing, coding and testing algorithms
K_{meta}	$0.55 \times \text{share of attempts without hints} + 0.45 \times (1 - \text{mean time ratio, capped at 1})$	Planning, monitoring and strategy selection
K_{ref}	$0.60 \times \text{mean reflection quality} + 0.40 \times \text{share of tasks corrected by the student after an error}$	Error analysis and improvement of one’s own solution
I	mean of the four indicators	Algorithmic competence overall

Note. Score equals 1 for a correct and 0 for an incorrect submission. Time ratio = solution time / (120 + 20 × task level) seconds. Reflection quality = min(1, words/35) plus 0.05 for each of eight strategy-related keywords present, capped at 1.

2.2. Generative model of synthetic learners

Each synthetic learner was given four latent traits (θ_{cog} , θ_{prac} , θ_{meta} , θ_{ref}), drawn from a multivariate normal distribution with unit variances and inter-correlations of .40, reflecting the expectation that the components are related but distinct. A fifth, construct-irrelevant trait, verbosity (v), was drawn independently from $N(0, 1)$. Behaviour on each task was generated as follows (Table 2). Task difficulty b was drawn from $N(0, 0.7^2)$, and the task level, used only for the normative time, from levels 3–7. The probability of requesting

a hint decreased with metacognition and overall ability and increased with difficulty. The probability of a correct submission depended on cognitive and practical traits, on hint use and, for repeated attempts, on the reflective trait, which represents the ability to analyse and correct one’s own errors. A student could submit up to three times per task. Solution time depended on difficulty, practical skill and metacognition. Reflection length depended on the reflective trait and on verbosity, while the number of strategy-related terms depended only on the reflective trait.

Table 2. Generative model used to simulate attempt logs (σ denotes the logistic function)

Behaviour	Model
Hint requested before the first submission	$P = \sigma(-0.5 - 0.9\theta_{meta} - 0.4(\theta_{cog} + \theta_{prac})/2 + 0.6b)$
Correct submission k ($k = 1, 2, 3$)	$P = \sigma(0.8\theta_{cog} + 0.8\theta_{prac} - b + 0.6 \cdot hint + 0.8 \cdot (k - 1) \cdot \sigma(\theta_{ref}))$
Time ratio	$\exp(0.35b - 0.35\theta_{prac} - 0.25\theta_{meta} + \epsilon), \epsilon \sim N(0, 0.35^2)$
Reflection length (words)	$\text{Poisson}(\max(1, 12 + 10\theta_{ref} + 8v))$
Strategy-related keywords (of 8)	$\text{Binomial}(8, \sigma(-1 + 1.2\theta_{ref}))$

The generated attempts were passed, in chronological order, to the scoring functions copied verbatim from the production code; only the database query was replaced by an in-memory list limited to the same window of attempts. All analyses used fixed random seeds, and the code is provided as supplementary material.

2.3. Analyses

Validity (RQ1, RQ2). For 3,000 learners with 40 tasks each, we computed the correlation of every indicator with every latent trait. To separate the contributions of correlated traits, each standardised indicator was regressed on the four standardised traits and verbosity; the standardised coefficients (β) show which characteristics drive the indicator. Correlations among the indicators were compared with the true trait inter-correlation of .40.

Reliability (RQ3). For windows of 10, 20, 40 and 80 attempts, two independent sets of tasks were generated for each of 2,000 learners with fixed traits, and the correlation between the two resulting indicator values was taken as parallel-forms reliability.

Robustness (RQ4). In each of two scenarios, 20% of 3,000

learners adopted a response strategy. In keyword padding, the student wrote 36-word reflections containing all eight strategy terms regardless of content. In copying, the student pasted a generated solution: first-attempt success with probability .95, no hints, solution time about one third of the norm and short reflections. Indicator values of these learners were compared with values computed for the same learners working genuinely, and the convergent correlations in the mixed population were compared with those in the genuine population.

3. RESULTS

3.1. Convergent and discriminant validity (RQ1)

All four indicators correlated substantially with their intended traits: $r = .79$ for K_{cog} , $.79$ for K_{prac} , $.87$ for K_{meta} and $.78$ for K_{ref} . The regression analysis, however, showed that convergence did not imply specificity (Table 3). K_{cog} and K_{prac} depended equally on the cognitive and the practical trait ($\beta = .57$ and $.56$ for both indicators), and the two indicators correlated at $r > .99$ with each other, far above the true trait correlation of .40. K_{meta} was driven mainly by metacognition ($\beta = .68$) but also by practical skill ($\beta = .38$). K_{ref} was driven

mainly by the reflective trait ($\beta = .86$) and was the most distinct indicator, correlating only .26–.28 with the others.

Table 3. Standardised regression coefficients of each indicator on the latent traits and verbosity (N = 3,000, window = 40 attempts)

Indicator	θ_{cog}	θ_{prac}	θ_{meta}	θ_{ref}	Verbosity	R^2	Convergent r
K_{cog}	.57	.56	−.08	.06	.00	.89	.79
K_{prac}	.57	.56	−.08	.06	.00	.89	.79
K_{meta}	.10	.38	.68	−.01	−.01	.90	.87
K_{ref}	−.10	−.10	.01	.86	.48	.86	.78

Note. Bold values mark the intended trait. Correlations among indicators: $K_{\text{cog}}\text{--}K_{\text{prac}} > .99$; $K_{\text{cog}}\text{--}K_{\text{meta}} = .64$; $K_{\text{prac}}\text{--}K_{\text{meta}} = .65$; K_{ref} with the others = .26–.28.

Two different mechanisms produce these results, and it is important to separate them. The near-identity of K_{cog} and K_{prac} is structural: because the score of a submission is binary and equal to its correctness, K_{cog} reduces to the share of correct submissions, and K_{prac} differs from it only by a small term for repeated attempts. This would hold under any generative model. By contrast, the equal dependence of both indicators on the cognitive and practical traits reflects our assumption that correctness depends equally on both; if one trait dominated correctness, both indicators would follow that trait. Similarly, the contribution of practical skill to K_{meta} arises because the time term rewards fast solutions, and fast solutions are produced by skilful students regardless of how they plan and monitor their work.

The indicators also differed in their distribution. Mean values were 0.56 (K_{cog}), 0.60 (K_{prac}), 0.43 (K_{meta}) and 0.41 (K_{ref}). With the fixed cut-offs, 26–27% of learners were classified as “high” on the cognitive and practical components but only 6% on the metacognitive and 3% on the reflective component, although all traits had the same distribution. The cut-offs are therefore

not comparable across components.

3.2. Construct-irrelevant variance (RQ2)

Verbosity, which by construction had nothing to do with competence, had no influence on the first three indicators but a substantial influence on K_{ref} ($\beta = .48$, $r = .48$). The source is the reflection-quality heuristic, in which length up to 35 words contributes most of the score. A student who habitually writes long answers therefore obtains a higher reflective indicator than an equally reflective student who writes briefly.

3.3. Reliability (RQ3)

Reliability increased with the window size for all indicators (Figure 1). With the default window of 40 attempts, parallel-forms reliability was .91 for K_{cog} and K_{prac} , .91 for K_{meta} , .96 for K_{ref} and .95 for the integral indicator. With only 10 attempts it fell to .74–.75 for the first three indicators. The reflective indicator was the most reliable at every window size, partly because verbosity is a stable characteristic of the learner: reliable variance is not necessarily valid variance.

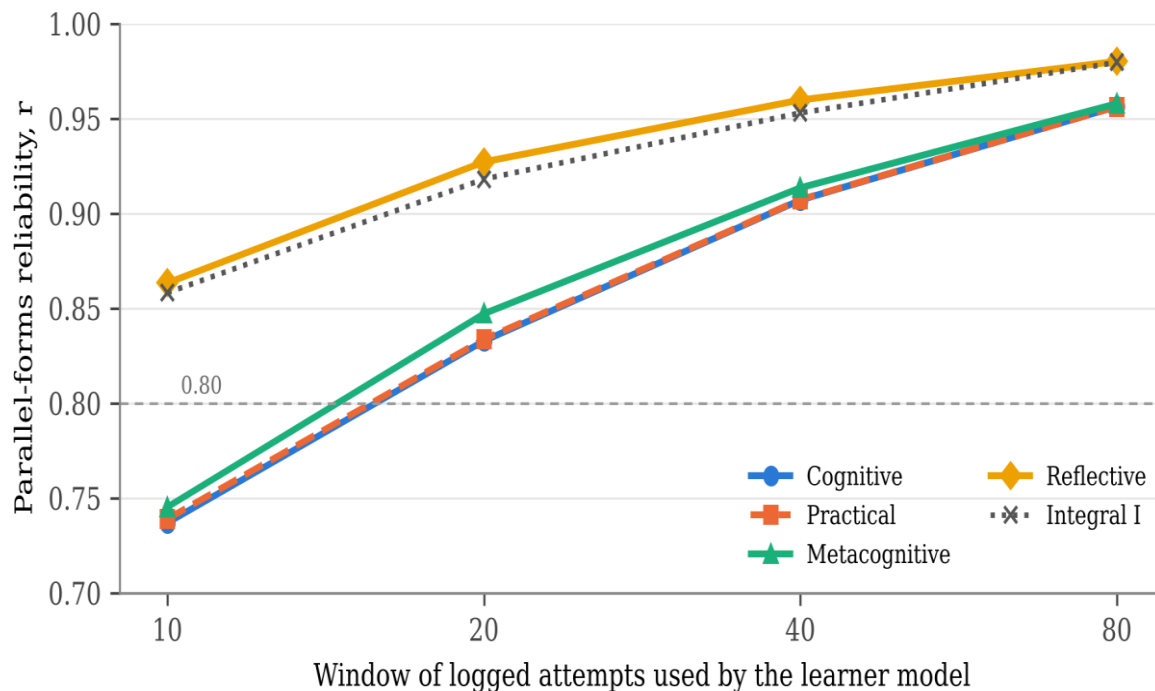


Figure 1. Parallel-forms reliability of the indicators as a function of the number of logged attempts used (2,000 synthetic learners per window). The cognitive and practical curves overlap.

3.4. Robustness to response strategies (RQ4)

Keyword padding affected only the reflective indicator but did so strongly (Table 4, Figure 2). For the padding learners, K_{ref} rose by 0.32 on average, and the share classified as “high” increased from 3% to 36%. In the mixed population, the correlation of K_{ref} with the reflective trait dropped from .78 to .55.

Copying generated solutions had a much broader effect. The

cognitive, practical and metacognitive indicators of copiers rose by 0.36–0.42, and **all** copiers were classified as “high” on these three components, compared with 26%, 28% and 6% of the same learners working genuinely. Their integral indicator rose from 0.50 to 0.72. Only the reflective indicator decreased, because copiers wrote short reflections and had no errors to correct. With 20% copiers in the group, the convergent correlations of all four indicators fell to .54–.56.

Table 4. Effect of two response strategies adopted by 20% of learners (N = 3,000)

Indicator	Padding: change mean	Padding: in “high”, %	Padding: convergent r	Copying: change mean	Copying: in “high”, %	Copying: convergent r
K_{cog}	0.00	26 → 25	.80 → .80	+0.39	26 → 100	.80 → .55
K_{prac}	0.00	28 → 26	.79 → .79	+0.36	28 → 100	.79 → .54
K_{meta}	0.00	6 → 6	.87 → .87	+0.42	6 → 100	.87 → .56
K_{ref}	+0.32	3 → 36	.78 → .55	−0.29	3 → 0	.78 → .56

Note. Changes in mean and “high” shares refer to the learners adopting the strategy, compared with the same learners working genuinely. Convergent r refers to the whole population, without and with the strategy.

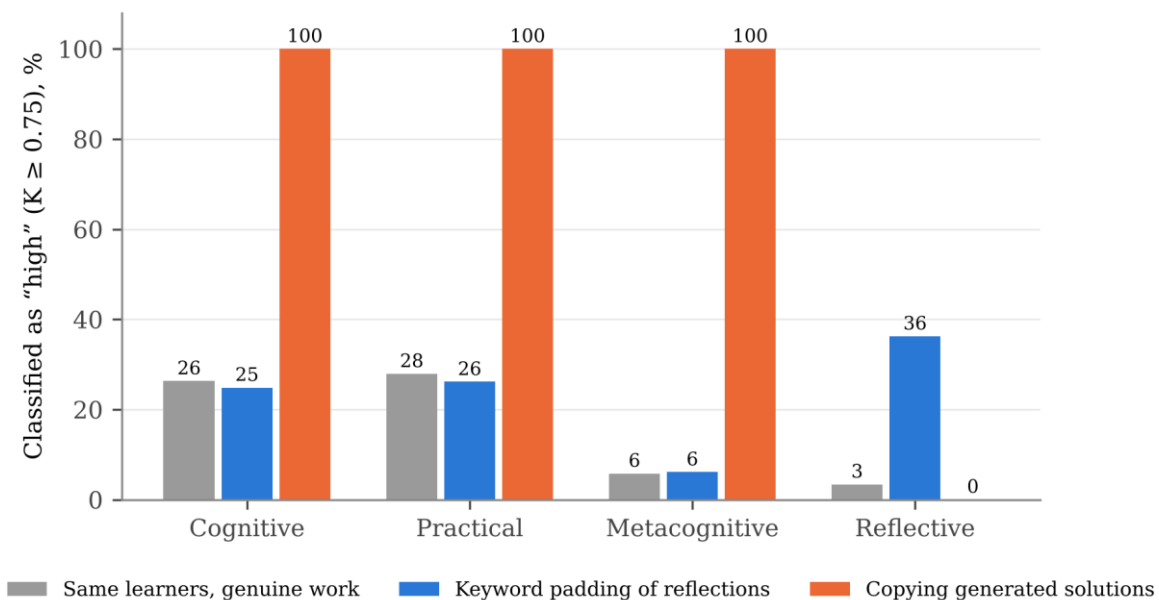


Figure 2. Share of learners classified as “high” ($K \geq 0.75$) when working genuinely and when padding reflections with keywords or copying generated solutions (the same 20% of learners in each condition).

4. DISCUSSION

This study asked whether the indicators of a process-based learner model measure what they claim before they are used to inform teachers and students. The answer is mixed. On the positive side, every indicator tracked its intended trait with correlations of .78–.87, and the default window of 40 attempts gave high reliability. On the negative side, the analysis revealed four problems that would not be visible from reliability or convergent correlations alone, and each points to a concrete revision.

Separate the evidence for the cognitive and practical components. Two indicators that correlate above .99 cannot inform a teacher about two different components; showing them separately on a dashboard suggests a distinction that the data do not support. In evidence-centred terms, the two claims need different observations (Mislevy et al., 2003). Conceptual knowledge can be observed in tasks that do not require writing code, such as predicting the output of a given program, tracing variables or explaining an algorithm (Lister et al., 2004), whereas practical skill is observed in code-writing tasks and in the quality of the testing and debugging process. The system already contains both short-answer and code tasks; computing K_{cog} only from the former and K_{prac} only from the latter, with partial credit for passed tests rather than a binary score, is a direct remedy.

Adjust time for skill. Rewarding fast solutions makes the metacognitive indicator partly a measure of practical skill. Metacognitive regulation is better reflected in how a student uses resources than in how quickly a task is solved (Zimmerman, 2002): requesting a hint after an unsuccessful

attempt rather than before trying, revising a plan after a failed test, or estimating difficulty accurately. Time can be retained if it is expressed relative to the time expected for a student of the same skill, which removes the share explained by practical ability.

Score reflections by content, not length. The length-based heuristic introduces construct-irrelevant variance and is easily gamed by keyword padding. Reflection quality should be judged with a short rubric (e.g., whether the student identifies the error, names its cause and states a strategy for the future), applied by teachers to a sample and, if automated, validated against teacher ratings. Large language models can assist with such scoring, but their judgements also require validation and teacher oversight (Hooshyar et al., 2025).

Design for generative AI. The copying scenario shows that process indicators based on correctness, speed and hint use reward exactly the trace that a pasted solution produces. Detection of AI-generated code is unreliable and is not recommended as a primary safeguard. A more robust approach is to collect evidence that a pasted solution does not provide: predicting and explaining program behaviour, finding and explaining errors in given code, constructing test cases, and short oral defence of submitted solutions (Denny et al., 2024; Prather et al., 2023). Anomalous patterns, such as fast first-attempt successes at a level far above previous performance, can be flagged for the teacher rather than scored automatically.

Calibrate the cut-offs. Identical cut-offs produced very different shares of “high” learners across components with identical trait distributions. Cut-offs should be set per

indicator, for example by standard-setting with teachers or by anchoring them to external criteria, before levels are reported.

More generally, the results illustrate a point made in the validity literature: reliability is necessary but not sufficient (Messick, 1995). The reflective indicator was the most reliable precisely because part of it measured a stable but irrelevant habit. Learning analytics dashboards that report such indicators risk giving teachers confident but misleading information, which is one reason why recent work calls for human-centred and responsible use of AI in education (Hooshyar et al., 2025).

4.1. Limitations

The results depend on the generative model. Its functional forms and parameters are plausible but not estimated from real data, and real students differ in ways the model omits, such as motivation, collaboration and changes in skill during the window. The structural findings—the near-identity of K_{cog} and K_{prac} , the dependence of K_{ref} on text length and the susceptibility of correctness- and speed-based indicators to copying—follow from the scoring rules and do not depend on these assumptions; the exact magnitudes do. The strategy scenarios are stylised, and the share of 20% was chosen for illustration. Finally, simulation can reveal threats to validity but cannot establish validity; that requires data from students, independent measures of the four components and teacher judgements.

4.2. Future work

The next step is to implement the proposed revisions in AlgoAdapt and to evaluate the revised indicators with first-year students, relating them to independent tests of algorithmic knowledge, rubric-scored code and reflection samples rated by teachers. Repeating the present simulation for the revised indicators will show whether the structural problems have been removed before the field study begins.

5. CONCLUSION

A process-based learner model that estimates cognitive, practical, metacognitive and reflective components of algorithmic competence from attempt logs was examined with simulated learners whose traits were known. The indicators were reliable and correlated with their target traits, yet two of them could not be distinguished, one was contaminated by practical skill, one by verbosity, and three were inflated to the maximum by copying generated solutions. Simulation-based checks of this kind are inexpensive and should precede the use of process indicators for reporting to teachers or for grading; the specific revisions proposed here—separate evidence sources, skill-adjusted time, rubric-based reflection scoring, AI-resilient evidence and calibrated cut-offs—apply to many adaptive programming environments beyond the system studied.

Declarations

Funding. This research received no specific grant from any funding agency.

Conflict of interest. The author is the developer and copyright holder of the AlgoAdapt software (certificate No. DGU 64245).

Ethics. The study used simulated data only and did not involve human participants.

Data and code availability. The simulation code and results are available as supplementary material.

Use of generative AI. [The author should describe here, according to the target journal's policy, whether and how AI tools were used in preparing the manuscript.]

REFERENCES

1. American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). Standards for educational and psychological testing. American Educational Research Association.
2. Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: When students “game the system”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 383–390). ACM. <https://doi.org/10.1145/985692.985741>
3. Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. *Communications of the ACM*, 67(2), 56–67. <https://doi.org/10.1145/3624720>
4. Feinberg, R. A., & Rubright, J. D. (2016). Conducting simulation studies in psychometrics. *Educational Measurement: Issues and Practice*, 35(2), 36–49. <https://doi.org/10.1111/emip.12111>
5. Hooshyar, D., Šir, G., Yang, Y., Kikas, E., Hämäläinen, R., Kärkkäinen, T., Gašević, D., & Azevedo, R. (2025). Towards responsible AI for education: Hybrid human-AI to confront the elephant in the room. *Computers and Education: Artificial Intelligence*, 9, 100524. <https://doi.org/10.1016/j.caeai.2025.100524>
6. Ihanola, P., Ahoniemi, T., Karavirta, V., & Seppälä, O. (2010). Review of recent systems for automatic assessment of programming assignments. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (pp. 86–93). ACM. <https://doi.org/10.1145/1930464.1930480>
7. Ihanola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C., & Toll, D. (2015). Educational data mining and learning analytics in programming:

Literature review and case studies. In Proceedings of the 2015 ITiCSE on Working Group Reports (pp. 41–63). ACM. <https://doi.org/10.1145/2858796.2858798>

8. Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., Seppälä, O., Simon, B., & Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4), 119–150. <https://doi.org/10.1145/1041624.1041673>
9. Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist*, 50(9), 741–749. <https://doi.org/10.1037/0003-066X.50.9.741>
10. Mislavy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*, 1(1), 3–62. https://doi.org/10.1207/S15366359MEA0101_02
11. Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38(11), 2074–2102. <https://doi.org/10.1002/sim.8086>
12. Pellegrino, J. W., Chudowsky, N., & Glaser, R. (Eds.). (2001). *Knowing what students know: The science and design of educational assessment*. National Academy Press.
13. Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluw, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Savelka, J. (2023). The robots are here: Navigating the generative AI revolution in computing education. In Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (pp. 108–159). ACM. <https://doi.org/10.1145/3623762.3633499>
14. Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>
15. Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30–40.
16. Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41(2), 64–70. https://doi.org/10.1207/s15430421tip4102_2