Practice as the Key to Success: Understanding the Role of Prior Knowledge, Affective States and Learning Resources in Computer Science Education

Andres Felipe Zambrano
University of Pennsylvania
Philadelphia, Pennsylvania, USA
azamb13@upenn.edu

Jiayi Zhang
University of Pennsylvania
Philadelphia, Pennsylvania, USA
joycez@upenn.edu

Maciej Pankiewicz
University of Pennsylvania
Philadelphia, Pennsylvania, USA
mpank@upenn.edu

Ryan S. Baker
Adelaide University
Adelaide, South Australia, Australia
ryanshaunbaker@gmail.com

Abstract

Introductory programming courses (CS1) bring together students with diverse prior experiences, which shape their use of learning resources, emotional responses, and academic performance. This study employs structural equation modeling, self-reported affective data, and multimodal interaction logs to investigate how prior programming knowledge affects affect, resource utilization, and outcomes in a CS1 course that features an automated assessment tool (AAT), instructional videos, and worked examples. Students who persisted with practice and advanced beyond basic tasks achieved the strongest outcomes, though they followed different emotional pathways. By contrast, relying solely on videos or worked examples did not significantly lead to success, and disengagement was generally tied to weaker performance. Novices often reported confusion and frustration; while these emotions sometimes hindered learning, they also drove deeper engagement with the AAT, improving outcomes for those who persisted. Experienced students, however, more often reported boredom, which consistently reduced practice and led to poorer outcomes. These findings underscore the need for adaptive support that balances challenge with guidance to sustain engagement and promote success for diverse learners in CS1.

CCS Concepts

• **Applied computing** → **Interactive learning environments**; • **Human-centered computing** → *Human computer interaction (HCI)*.

Keywords

Computer Science Education, Programming Learning, Affective States, Automated Assessment Tools, Engagement, Sustained Practice

LAK 2026, Bergen, Norway

ACM ISBN 979-8-4007-2066-6/2026/04
https://doi.org/10.1145/3785022.3785032

ACM Reference Format:
Andres Felipe Zambrano, Jiayi Zhang, Maciej Pankiewicz, and Ryan S. Baker. 2026. Practice as the Key to Success: Understanding the Role of Prior Knowledge, Affective States and Learning Resources in Computer Science Education. In *LAK26: 16th International Learning Analytics and Knowledge Conference (LAK 2026), April 27-May 01, 2026, Bergen, Norway.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3785022.3785032

1 Introduction

Introductory programming courses (commonly referred to as CS1) form a critical foundation in the first year of college for students studying computer science and related fields, as they develop the computational thinking skills needed for advanced coursework and future careers [24, 33]. Students enter these courses with programming backgrounds that range from no prior exposure to extensive experience through coursework, extracurricular activities, or self-study. Prior experience has been widely associated with success in CS1 [13, 41, 42], shaping students' perceptions and motivations, which are themselves crucial for long-term achievement [33].

To accommodate students' diverse backgrounds, CS1 classes employ a range of learning resources, such as automated assessment tools (AATs; e.g., [16, 30, 31]), instructional videos [39], and worked examples [34], that allow students to engage with content in different ways. While these resources are generally beneficial, they do not always produce the intended outcomes. For example, although AATs enable students to practice programming while offering immediate feedback, both critical for developing skills and understanding foundational concepts [8, 31], some students still find the feedback unhelpful for correcting mistakes or supporting learning [30]. Similarly, while interactive videos often support learning, their effectiveness depends on both video design and the strategies students adopt when engaging with them, with benefits differing depending on students' prior knowledge [1, 17, 39]. Worked examples can also scaffold problem-solving and code writing for students with previous programming experience [34], but further research is needed to determine their effectiveness for novices.

The connection between engagement behaviors and learning outcomes may be mediated by students' prior knowledge and the affective states triggered by their perceived level of challenge. According to several affective frameworks [9, 23, 29], experienced students may disengage out of boredom when the material feels trivial or overly familiar. Conversely, novices who repeatedly struggle with errors in AATs may experience confusion or frustration, which can lead them to view feedback as unhelpful and disengage from the platform, or to rely on trial-and-error strategies rather than deliberate reflection to address misunderstandings. Both confusion and boredom have been linked to adverse outcomes, whereas engaged concentration during coding and reading has been linked with higher achievement [6, 32]. Although prior knowledge, affective states, and behaviors have each been connected to learning outcomes, their interactions remain less understood. Identifying these mechanisms through which they influence each other is essential for designing learning supports that address the diverse needs of students.

This study examines how these three dimensions relate to students' learning outcomes in a CS1 course that employs AATs, instructional videos, and worked examples. Specifically, we ask: How does prior programming experience shape students' affective experiences and use of learning resources, and how do these factors predict final test performance? By analyzing self-reported affective states and patterns of resource use, we aim to elucidate the mechanisms by which prior knowledge influences learning trajectories in CS1. Ultimately, this work aims to inform the design of more widely effective supports for students in introductory programming courses.

2 Related Work

2.1 Prior Knowledge and Affective States as Mediators of Behavior and Learning

Prior programming experience has been shown to influence students' performance and engagement with course activities. Bui et al. [7] found an achievement gap between students with no prior experience and those with some programming background, along with differences in self-efficacy and retention, suggesting that prior knowledge shapes not only academic outcomes but also confidence. Similarly, Wilcox and Lionelle [41] observed that students with stronger prior knowledge initially outperformed novices on exams and quizzes in CS1. However, by the end of CS2, this performance gap had closed, with no significant differences in overall scores. Hagan and Markham [13] likewise reported higher CS1 assessment scores for students with prior experience, regardless of the language learned, though final test scores showed only a nonsignificant positive effect. One explanation is that experienced students may find introductory concepts trivial, leading to disengagement that limits learning. Supporting this, Veerasamy et al. [38] found that while experienced students ultimately earned the highest final exam scores, they often paid less attention in lectures because much of the material felt familiar.

Several theoretical models, such as Csikszentmihalyi's [9] flow theory, Pekrun's [29] Control-Value Theory (CVT), D'Mello and Graesser's [11] model of Affective Dynamics, and Ocumpaugh et al.'s [23] Skills, Difficulty, Value, Efficacy, and Time (SDVET) model, explain how affective experiences, shaped by students' prior knowledge and perceived task challenge, influence behavior and learning. Although these models differ in how they emphasize motivational constructs such as perceived value, control, or self-efficacy, they converge on the idea that both excessive difficulty (often faced by novices) and excessive ease (often faced by more experienced students) can lead to disengagement. Overly complex tasks may provoke confusion or frustration that, if unresolved, devolves into withdrawal, while trivial tasks may generate boredom when students see little opportunity for new learning.

Empirical studies support these models in CS1 contexts. Rodrigo et al. [32] observed that negative affective states, such as confusion and boredom, were negatively associated with achievement in a CS1 class, while engaged concentration and on-task behaviors, possibly linked to such concentration, were positively associated with achievement. Similarly, Bosch and D'Mello [6] found that experiencing confusion and boredom while practicing programming tasks was negatively correlated with learning outcomes. In contrast, transitions between confusion and frustration, and from boredom to concentration (a potential signal of reengagement) were positively associated with improved performance. Their results also revealed direct connections between affect and behavior. Confusion or frustration often preceded hint usage, while engagement and curiosity were associated with reading and coding.

Affective states also shape how students interact with instructional videos. A self-report study by Aydin et al. [3] found that pausing and seeking backwards information commonly occurred when students felt confused, suggesting attempts to clarify content. Seeking forward, by contrast, was often driven by boredom and a desire to skip familiar or uninteresting segments. Boredom was also frequently triggered by dense or poorly delivered explanations, sometimes leading to early termination of video viewing. In contrast, engagement was associated with continuous viewing, particularly when topics were seen as relevant or aligned with hands-on practice.

These findings indicate that prior knowledge and affective experiences are critical dimensions for understanding learning processes in CS1. However, while prior research provides strong evidence that emotions shape engagement and outcomes, and suggests plausible ways in which prior knowledge might moderate these dynamics, further work is needed to establish how affective experiences systematically differ across levels of prior programming knowledge and impact on behaviors and learning.

2.2 Engagement Behaviors as Predictors of Outcomes

Students' engagement with multiple resources is central to their learning outcomes. Research on AATs highlights several benefits, including opportunities for repeated practice and immediate feedback [8, 31]. These features are particularly valuable because they strengthen both understanding and confidence in programming, which is especially important since insufficient practice is a common barrier to learning the subject [18]. Pettit et al. [30] likewise found that students valued AATs for their immediate, objective assessment and supportive feedback. However, some students reported frustration with rigid evaluation criteria and feedback that was perceived as being less helpful than feedback from a human instructor, which in turn discouraged engagement.

Instructional videos have also been shown to improve learning in higher education when designed according to multimedia principles such as coherence and segmentation [22, 37]. In STEM contexts, Adler et al. [1] found that videos disproportionately benefited students with higher prior knowledge, even though attention levels were similar across groups. In computer science education, Wan et al. [39] showed that scaffolding embedded in instructional videos enhanced students' computational thinking and programming performance compared to unscaffolded videos. Wang et al. [40] likewise found a positive association between video use and coding practice, suggesting that students often combine resources in complementary ways. MacHardy and Pardos [21] further observed, using knowledge tracing models, that while some videos added predictive value to mastery, others did not contribute meaningfully to learning. Finally, Kuhlmann et al. [17] found that behaviors such as speed-watching and rewinding positively predicted exam scores, particularly among students with lower prior knowledge.

Readings and examples also provide structured guidance that supports problem-solving and programming learning across different levels of expertise. Sankaranarayanan et al. [34] found that advanced CS students learning new concepts benefited more from worked example-based reflections than from unguided problem-solving, suggesting that these materials remain valuable even beyond the novice stage. However, it is still unclear whether students without prior programming experience can benefit equally from this type of resource.

This body of research shows that engagement with AATs, videos, and worked examples is an important predictor of learning outcomes. However, the benefits of each resource depend not only on its design but also on how students use it. Engagement patterns and the overall impact of these resources are shaped by prior knowledge and its connection with affective states. This literature highlights the need to study engagement in relation to both prior knowledge and affect, as their interaction likely shapes students' learning in CS1 but remains underexplored.

3 Methods

3.1 Data Context

In this study, we analyze data from 236 undergraduate computer science students enrolled in a mandatory first-semester introductory programming course (CS1) during the Fall 2024-2025 academic year at a large European university. Informed consent was obtained from all participants. The course was taught through theoretical lectures and practical laboratory sessions, both occurring on a weekly basis. Students were provided with multiple opportunities to engage with core programming concepts through videos, code examples, and programming assignments for self-practice. An AAT allowed students to submit code for immediate feedback on correctness and errors, facilitating hands-on practice with foundational programming topics. It offered access to 229 coding tasks, categorized into the following: 10 introductory tasks, 50 on types and variables, 32 on conditional statements, 34 on recursion, 35 on arrays, 8 that combined both recursion and arrays, 26 on loops, and 44 that combined loops and arrays. 32 instructional videos were distributed across the same key topics: 5 introductory videos, 5 on types and variables, 7 on conditional statements, 5 on arrays, 3 on recursion, 3 on recursion with arrays, and 4 on loops and arrays. Complementing this platform, a learning management system provided access to coding examples. Usage of all provided resources was optional for students and did not influence the final grade, although the instructor encouraged the use of all the resources throughout the semester.

During the first classes, students completed a survey to rate their prior programming experience. They were asked: "On the scale 1–5, where 1 is zero experience, and 5 is a lot of experience, please rate your basic programming knowledge (types, variables, conditional statements, recursion, loops, arrays)." During this initial phase, students also completed a pre-test to assess their baseline knowledge, complementing their self-evaluation of their prior expertise; the pre-test and self-evaluation were strongly correlated ($\rho = 0.72$, $p < 0.001$). A final test was administered at the end of the semester featuring code-based questions. Both tests were standardized to percent correct to facilitate their comparison. Additionally, throughout the semester, affective data were collected via self-reports in a survey triggered with a 10% probability after each submission on the automated assessment platform. Students were prompted to indicate their affective state, choosing among Anxiety, Boredom, Concentration, Confusion, Frustration, and Other, selected because of their observed importance for learning [15].

3.2 Structural Equation Modelling

We employed Structural Equation Modelling (SEM) to examine how students' pre-test scores influenced their access to learning resources and affective experiences throughout the semester, and how these factors, in turn, predicted final test performance. The model includes one exogenous predictor (pre-test score), two sets of endogenous mediators—affective states (boredom, confusion, frustration, engaged concentration, and anxiety) and resource access behaviors (LMS activity, video watching, and submissions in the automated assessment tool)—and one endogenous outcome variable (post-test score). We assumed a temporal and causal ordering in which prior knowledge influence affective states, affective states influence resource access behaviors, and these behaviors ultimately predict the final outcome.

Our affective data came from student self-reports, which depended on voluntary submissions. As such, some students seldom or never reported on their affect, producing a possible missing-not-at-random (MNAR) scenario [19]. Several distributions, especially those for less common affective states such as boredom or anxiety, where high proportions of students reporting zero occurrences, likely due to the low engagement of some students with the automated assessment platform, resulting in little or no self-reported data. This phenomenon posed challenges for SEM and other linear regression-based methods, where such skew can strongly bias estimates. To mitigate this issue, we excluded students who submitted fewer than 10 affect reports, as their data were insufficient to reliably characterize their affective states. Including these students risked misrepresenting their experience, for example, suggesting they never experienced boredom when they may simply have had no opportunity to report it during the few times they responded. This decision reduced our dataset by approximately 25% and, given the MNAR nature of the missing data, limited the sample to students who were more engaged over the semester. However, this allowed us to focus on participants for whom we could more accurately estimate overall affective tendencies and their relationships with behaviors and learning outcomes. This focus also helped disentangle effects arising from genuine, sustained behavioral patterns from those caused purely by general disengagement, a factor widely recognized as detrimental to learning [12] but not central to the main research question of this study.

The proportion of zero responses for a specific category was substantially reduced after removing these students, but some variables still showed slightly skewed, zero-inflated distributions. To address this, we applied a rank transformation to minimize potential issues from non-normality. All regressions were checked for residual normality and homoskedasticity. We also assessed the monotonicity of all identified associations using generalized additive model (GAM) plots. All significant paths discussed in this paper showed linear relationships between the rank-transformed variables (monotonic associations).

To develop a parsimonious SEM, we began with an exploratory analysis using pairwise Spearman correlations rather than including all possible paths and variables. Only variables showing marginally or statistically significant correlations were retained, along with the specific paths reflecting those associations. This strategy reduced the risk of multicollinearity, preserved statistical power by limiting the number of estimated parameters, and improved model convergence and stability, considerations that were particularly important given the zero-inflated distributions observed in some mediators. It also minimized the likelihood of suppressor effects and spurious paths that can occur when unrelated variables are included. No correction for multiple comparisons was applied, as the correlations were used solely as an exploratory tool to inform the SEM rather than for formal statistical inference.

After identifying marginally and statistically significant paths in the initial model, we specified a reduced model that retained only these relationships, reflecting what we believe to be the underlying structure of the data. We evaluated the fit of this simplified model using standard indices—Comparative Fit Index (*CFI*) and Standardized Root Mean Square Residual (*SRMR*)—and compared its R^2 values to those of the initial, more saturated model. The reduced model demonstrated comparable explanatory power, supporting the adequacy and interpretability of the streamlined structure. Model parameters were estimated using a robust maximum likelihood estimator in Lavaan (version 0.6.19) in R. This estimator produces a scaled test statistic that adjusts for non-normality in the observed data. Following model estimation, residuals for all endogenous variables were examined, and no violations of homoscedasticity or residual normality assumptions were detected.

3.3 Secondary Analysis of Access to Each Type of Resource

In the primary analysis, we limited the number of predictors in the SEM to avoid substantially increasing the number of estimated parameters, which could compromise model stability and interpretability. However, we acknowledge that different types of video access and submission behaviors may have distinct effects on learning outcomes. Therefore, as a secondary analysis, we examined more granular patterns of learning resource usage, focusing on those specific resources that showed significant associations with post-test scores, while also accounting for students' prior expertise. To do so, we first categorized students into novice and experienced groups based on their responses to the initial prior-experience survey. Students who responded with a 1 or 2 on the Likert scale were classified as novices, while those who responded with a 3 or higher were categorized as experienced. This classification approach has been employed in previous studies (e.g., [28]). Within each prior experience group, students were further classified as high performers (top quartile of post-test scores), low performers (bottom quartile), and mid performers (second and third quartiles). We then conducted Mann-Whitney U tests to compare high and low performers within each experience group. Mid-performers were excluded due to the potential risk of spurious effects, as this group is likely heterogeneous, comprising students with varying levels of engagement and ability, which may obscure the contrasts of interest between the highest- and lowest-performing students.

For the automated assessment platform, we extracted and compared several features reflecting students' platform activity between the groups of interest. These features included the total number of solved tasks, the percentage of submissions containing compiler errors and non-compiler errors, and the distribution of submissions across different programming topics (e.g., Arrays, Conditional Statements, Loops, among others). We applied a Benjamini-Hochberg correction to control the false discovery rate. The same analysis was conducted for the other two learning resources.

Lastly, to examine temporal patterns and group differences in resource use over the semester, we compared weekly activity across performance and prior-experience groups. For each week, we calculated the average number of submissions, videos accessed, and examples viewed per student in each group. To capture broader patterns of (in)activity, including possible signs of procrastination, we also computed the total minutes of activity per week. Periods of more than 30 minutes between consecutive actions were treated as breaks and excluded from this total.

4 Results

4.1 Correlation Mining

Table 1 presents the pairwise correlations between all variables considered for the SEM. Final test scores were positively correlated with pre-test scores ($\rho = 0.40$, $p < 0.001$), the number of submissions made by students ($\rho = 0.12$, $p = 0.077$, marginal), and the proportion of self-reported boredom ($\rho = 0.15$, $p = 0.076$, marginal). The positive association between boredom and final test scores, although seemingly counterintuitive, can be explained in this case by the strong positive correlation between boredom and pre-test scores ($\rho = 0.28$, $p < 0.001$). This interpretation was supported by the very low and non-significant semipartial correlation of 0.012 between boredom and final test scores ($p = 0.871$) after controlling for the pre-test scores, suggesting that students with higher prior knowledge tended to report greater boredom, and that same higher prior knowledge also made them more likely to achieve higher final test scores. In contrast, the use of videos ($\rho = -0.24$, $p < 0.001$) and the self-reported experiences of confusion ($\rho = -0.18$, $p = 0.032$) and frustration ($\rho = -0.15$, $p = 0.072$, marginal) were negatively associated with final test scores. All of these relationships were subsequently incorporated into the regression model predicting final test performance.

Table 1: Pairwise Spearman correlation coefficients. Statistically significant and marginally significant associations are shown in bold.

	Post-test	Pre-test	Submissions	Examples	Videos	Anxiety	Boredom	Focus	Confusion	Frustration
Post-test	1.00	0.40*	0.12*	-0.03	-0.24**	-0.01	0.15*	0.08	-0.18**	-0.15*
Pre-test	0.40**	1.00	-0.30**	-0.24**	-0.49**	-0.17**	0.28**	-0.03	-0.24**	-0.12
Submissions	0.12*	-0.30**	1.00	0.22**	0.50**	0.09	-0.16*	0.04	0.01	0.16*
Examples	-0.03	-0.24**	0.22**	1.00	0.36**	0.09	0.10	-0.12	0.04	0.25**
Videos	-0.24**	-0.49**	0.50**	0.36**	1.00	0.05	-0.07	0.05	0.09	0.10
Anxiety	-0.01	-0.17**	0.09	0.09	0.05	1.00	-0.02	-0.42**	0.46**	0.28**
Boredom	0.15*	0.28**	-0.16*	0.10	-0.07	-0.02	1.00	-0.43**	0.05	0.25**
Focus	0.08	-0.03	0.04	-0.12	0.05	-0.42**	-0.43**	1.00	-0.43**	-0.64**
Confusion	-0.18**	-0.24**	0.01	0.04	0.09	0.46**	0.05	-0.43**	1.00	0.32**
Frustration	-0.15*	-0.12	0.16*	0.25**	0.10	0.28**	0.25**	-0.64**	0.32**	1.00

Note. * $p < 0.05$, ** $p < 0.01$.

Beyond its relationship with post-test performance and boredom, the pre-test score was also negatively correlated with examples ($\rho = -0.24$, $p < 0.001$) and video access ($\rho = -0.49$, $p < 0.001$), and the proportions of reported anxiety ($\rho = -0.17$, $p = 0.043$) and confusion ($\rho = -0.24$, $p = 0.003$). These associations led us to include pre-test scores as a predictor for all variables except focus and frustration, for which no associations were observed. The number of submissions was positively associated with the number of worked examples accessed through the LMS ($\rho = 0.22$, $p < 0.001$) and the number of videos watched ($\rho = 0.50$, $p < 0.001$), prompting us to include covariates among these three behavioral variables in the SEM. Submissions were also negatively correlated with boredom ($\rho = -0.16$, $p = 0.055$, marginally) and positively correlated with frustration ($\rho = 0.16$, $p = 0.058$, marginally). Accordingly, these two affective states, were included in the regression model predicting the number of submissions.

In the case of video usage, no significant correlations were found with any affective states. Therefore, the only variable included in the regression model predicting video access was the pre-test score. In contrast, examples access showed a positive association with frustration ($\rho = 0.25$, $p = 0.002$), which led us to include frustration, along with pre-test scores, in the corresponding regression model. As noted earlier, pre-test scores were correlated with all affective states except for focus and frustration, making these the only affective variables without a regression model that included pre-test as a predictor.

The analysis also revealed several significant correlations among the proportions of self-reported affective states. Focus was negatively correlated with all negatively valenced affective states (ρ range = -0.64 to -0.42, $p < 0.001$), a pattern consistent with theoretical expectations (e.g., [11]). Positive associations were observed among anxiety, confusion, and frustration (ρ range = 0.28 to 0.46, $p < 0.001$), a cluster of relationships previously theorized and documented in the literature [10]. Boredom was positively correlated only with frustration ($\rho = 0.25$, $p = 0.002$), but not with confusion or anxiety, aligning with models of affect transitions, which propose that frustration often transitions into boredom [11]. These observed interrelations among affective variables informed our inclusion of specific covariates between them in the SEM.

4.2 Structural Equation Modeling Analysis

We initially fit a SEM including all significant and marginally significant correlations identified in the correlation mining (Table 1). This model explained 27.8% of the variance in students' final test scores and yielded a SRMR of 0.042. Within this model, video access ($\beta = -0.13$, $p = 0.121$), boredom ($\beta = 0.11$, $p = 0.192$), and confusion ($\beta = -0.02$, $p = 0.780$) were not significant predictors of final test scores when pre-test scores, submissions, and frustration levels were also included. Similarly, the direct effect of pre-test scores on submissions was no longer significant ($\beta = -0.06$, $p = 0.192$) and instead operated indirectly through boredom (see final model in Figure 1). Finally, pre-test scores no longer significantly predicted anxiety ($\beta = -0.13$, $p = 0.121$).

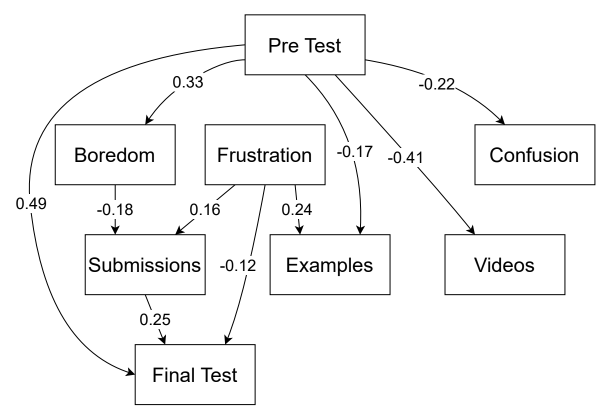

Figure 1: Final SEM retaining only significant paths.

After identifying the significant paths, we fit a more parsimonious model that only included them to evaluate whether the structural relationships derived from the initial model meaningfully represented the data. This reduced model (Figure 1) explained 26.7% of the variance in final test scores, similar to the 27.8% explained by the original model. The model had a *CFI* of 1.00 and an *SRMR* of 0.043, both within commonly accepted thresholds for adequate fit

Table 2: Pairwise Spearman correlation coefficients. Statistically significant and marginally significant associations are shown in bold.

Outcome	Predictor	Estimate	Std. Error	z	p-val
Final Test	Pre Test	0.49	0.07	6.92	< 0.001
Final Test	Submissions	0.25	0.12	2.19	0.029
Final Test	Frustration	-0.12	0.07	-1.81	0.071
Submissions	Boredom	-0.18	0.05	-3.33	0.001
Submissions	Frustration	0.16	0.05	2.97	0.003
Videos	Pre Test	-0.41	0.07	-5.59	< 0.001
Examples	Pre Test	-0.17	0.08	-2.16	0.031
Examples	Frustration	0.24	0.08	3.14	0.002
Boredom	Pre Test	0.33	0.08	4.35	< 0.001
Confusion	Pre Test	-0.22	0.08	-2.64	0.008

($CFI \geq 0.95$, $SRMR \leq 0.08$; [14]). All retained paths remained statistically significant or at least marginally significant (see Table 2).

The final model shows that pre-test scores positively predicted final test performance ($\beta = 0.49$, $p < 0.001$), an expected finding given that more experienced students begin the semester in a stronger position. Prior knowledge was also positively associated with the proportion of self-reported boredom ($\beta = 0.33$, $p < 0.001$), which, in turn, negatively predicted the number of code submissions during the semester ($\beta = -0.18$, $p = 0.001$). In addition, pre-test scores negatively predicted the number of video views ($\beta = -0.41$, $p < 0.001$), the code examples accessed ($\beta = -0.17$, $p = 0.031$), and confusion ($\beta = -0.22$, $p = 0.008$). These results suggest that students with higher prior knowledge were less likely to engage with the learning resources provided during the semester, likely because they perceived themselves as already proficient in the covered topics, experiencing less confusion with the content but more boredom, which reduced the number of submissions.

Among the affective states, only frustration (unrelated to prior knowledge) positively predicted both the number of submissions made by students ($\beta = 0.16$, $p = 0.003$) and their access to worked examples ($\beta = 0.24$, $p = 0.002$). However, of these learning resources, only the number of submissions significantly predicted final test performance when pre-test scores were also included in the regression model ($\beta = 0.25$, $p = 0.029$). Notably, while frustration positively predicted the number of submissions, it had a marginal negative association with final test scores ($\beta = -0.12$, $p = 0.071$). This pattern suggests that frustration may be beneficial if it motivates students to continue practicing, but beyond that potential benefit, it could still have a detrimental impact on learning (or that its signaling role about poor understanding is stronger than its role in motivating students to continue practicing).

4.3 Comparison Across Prior Knowledge and Performance Groups

To complement our primary analysis on the effects of practicing with the AAT, which emerged as the only resource that significantly predicted the final performance, we conducted a follow-up analysis examining more granular patterns of behavior. Specifically, we investigated whether the subjects of tasks students attempted to solve and the types of errors they commonly made differed across performance groups, while accounting for differences in prior expertise.

A comparison between high- and low-performing novices (final test scores of 85.2% and 26.4%, $p < 0.001$) revealed significant differences in their initial self-reported skill levels (mean = 1.73 vs. 1.41, $p = 0.013$; see Table 3), as well as in their prior knowledge (mean = 29.3% of correct answers vs. 16.5%, $p = 0.023$). These findings suggest that, among novices, students who enter the course with slightly higher prior knowledge may be more likely to engage in behaviors that support improved learning outcomes.

The analysis of submission behaviors revealed that novices with high performance completed more tasks (mean = 169.0 vs. 105.4, $p = 0.005$) and submitted more solution attempts (mean = 529.7 vs. 357.2, $p = 0.026$) than their low-performing peers. At first glance, this seems to contradict the negative association between prior knowledge and submission frequency observed in the SEM. However, this pattern likely reflects that students with low (but still some) prior knowledge are more willing or able to persist through more advanced tasks, resulting in a greater number of submissions. Novices also tended to submit code more frequently than any performance group within the experienced students, suggesting that the relationship between prior knowledge and submission behavior shifts depending on whether it is examined within or across prior experience groups.

High-performing novices also persisted more and continued to work with more advanced programming subjects, such as Recursion (18.2% of their submissions corresponded to these tasks, compared to only 9.8% for low performers, $p = 0.005$) and tasks combining Loops and Arrays (14.1% vs. 5.9%, $p = 0.004$). In contrast, low performers did not continue beyond Introductory tasks (24.4% vs. only 6.7%, $p = 0.011$). Additionally, the percentage of submissions containing syntax errors was higher for low performers (35.2% vs. 30.0%, $p = 0.013$). Meanwhile, high performers showed a higher rate of failed unit tests (29.2% vs. 19.9%, $p = 0.007$), likely reflecting the greater difficulty of the advanced tasks they attempted, being more prone to logical errors rather than syntax-related issues. These patterns suggest that difficulties with compiler errors, likely stemming from limited prior experience, may have prevented low-performing novices from progressing to more advanced tasks. As a result, they had fewer opportunities to practice core programming concepts

Table 3: Comparison between high- and low-performing novice and experienced students. Significant differences after the Benjamini-Hochberg correction are shown in bold.

Feature	Novices			Experienced		
	High (N=33)	Low (N=27)	p-val	High (N=56)	Low (N=35)	p-val
Reported Skill	1.73	1.41	0.013	3.88	3.46	0.012
% Pre Test	29.3	16.5	0.023	69.9	47.7	< 0.001
% Final Test	85.2	26.4	< 0.001	96.7	35.7	< 0.001
Submissions	529.70	357.22	0.026	360.18	256.40	0.097
Videos	18.67	24.04	0.671	7.41	11.46	0.018
% Watched per Video	46.1	48.3	0.613	32.5	40.8	0.242
Repeated Videos	4.27	6.04	0.348	1.48	2.31	0.009
LMS	47.09	46.63	0.911	29.54	39.77	0.957
Total Correct Tasks	169.03	105.44	0.005	152.61	92.91	0.003
% Correct Submissions	39.2	43.8	0.494	52.9	51.2	1.000
% Sub. Compiler Errors	30.0	35.2	0.013	19.8	25.0	0.018
% Sub. Failed Unit Tests	29.2	19.9	0.007	26.6	22.9	0.031
% Sub. Introduction	6.7	24.4	0.011	11.0	20.0	< 0.001
% Sub. Types	23.6	32.6	0.174	18.1	31.1	0.005
% Sub. Conditionals	10.9	12.5	0.814	8.2	10.0	0.778
% Sub. Arrays	11.2	6.5	0.057	7.1	8.2	0.672
% Sub. Loops	9.2	5.4	0.043	9.3	4.0	0.001
% Sub. Loops-Arrays	14.1	5.9	0.004	22.8	6.8	< 0.001
% Sub. Recursion	18.2	9.8	0.005	17.1	17.9	1.000
% Sub. Recursion-Arrays	6.0	2.9	0.021	6.4	1.9	< 0.001

before the final test, which may have contributed to their lower performance. In contrast, high-performing novices persisted to more advanced tasks where failed unit tests became more common. This likely provided them with opportunities to practice debugging and develop logical reasoning skills beyond syntax correction, which may have supported their higher learning outcomes.

As observed with novice students, experienced students who entered the course with higher self-reported skill levels (means: 3.88 vs. 3.46, $p = 0.012$; see Table 3) and greater prior knowledge (means = 69.9% vs. 47.7%, $p < 0.001$) achieved significantly higher final test scores (means = 96.7% vs. 35.7%, $p < 0.001$). Although high-performing experienced students tended to make more submissions than their low-performing counterparts, this difference did not reach statistical significance (means = 360.2 vs. 256.4, $p = 0.097$). However, high performers solved significantly more tasks overall (means = 152.6 vs. 92.9, $p = 0.003$) and submitted proportionally more solution attempts in advanced subjects, including Loops (9.3% vs. 4.0%, $p = 0.001$), combined Loops and Arrays (22.8% vs. 6.8%, $p < 0.001$), and combined Recursion and Arrays (6.4% vs. 1.9%, $p < 0.001$). In contrast, low-performing experienced students did not advance beyond Introductory tasks (20.0% vs. 11.0%, $p < 0.001$) and Variable Types tasks (31.1% vs. 18.1%, $p = 0.005$), also having a higher proportion of submissions with compiler errors (25.0% vs. 19.8%, $p = 0.018$).

Notably, although low-performing experienced students made more syntax-related errors than their high-performing peers, their overall proportion of submissions with compiler errors (25%) was still lower than that of both high and low performing novices (30 and 35%, respectively). This suggests that their reduced engagement with more advanced practice tasks is unlikely to be due to difficulties with basic syntax during introductory tasks.

Despite beginning the course with substantially higher levels of prior knowledge and self-reported expertise, their final test scores (36%) were substantially lower than those of high-performing novices (85%). These results suggest that their low performance on the final test may not be due to a lack of prior knowledge, but rather to limited engagement with the automated assessment platform, particularly with advanced tasks. It is possible that these students, believing they had already mastered the early material, disengaged prematurely before encountering content that would challenge and further develop their skills. In this sense, their disengagement may reflect a lack of challenge, rather than the excessive challenge that appeared to inhibit low-performing novices.

Overall, these findings suggest that while prior knowledge is an important predictor of academic performance, it is not sufficient on its own. Progressing from basic introductory tasks to more advanced coding exercises, particularly those that promote debugging skills and logical reasoning, appears to play a more critical role in determining final outcomes in this course.

We replicated this analysis for video and LMS access by topic and found no differences in the distribution of worked examples or videos accessed by topic (e.g., approximately 20% were introductory videos and about 15% focused on conditional statements across all groups). We also observed no differences in the average percentage of video watched across groups with similar prior knowledge. Students in both novice groups tended to watch about 50% of the videos they accessed, whereas students in both experienced groups tended to watch around 35%. The only significant findings in this secondary analysis were that low-performing experienced students tended to watch more videos than high-performing experienced students (11.46 vs. 7.41, $p = 0.018$) and also tended to rewatch more videos (2.31 vs. 1.48, $p = 0.009$). These results suggest that practicing programming skills (and progressing through increasingly challenging and advanced tasks) may be more important for student learning than relying solely on videos or worked examples.

4.4 Temporal Analysis

Figures 2A- 2C show the average weekly engagement of each group with different learning resources. Figure 2D displays the average weekly minutes of activity per group. Both high-performing groups (novice and experienced) showed an initial peak in submissions between weeks 3 and 6. However, they primarily engaged with the AAT, making more submissions and spending more time on it during the second half of the semester, particularly in the month leading up to the final test (weeks 12-16; Figure 2A). In contrast, the low-performing groups did not exhibit this pronounced increase in practice before the final test and appeared disengaged for most of the semester, aside from small early peaks around week 4.

In terms of video access (Figure 2B), the low-performing groups, for both experienced and novice students, showed a pronounced peak in the first weeks of the semester. This pattern suggests that these students, in some cases, may have preferred to watch videos to learn the content before beginning to practice with the AAT. For example, the peak in video views for low-performing novices occurred in week 3, immediately before their peak in submissions in week 4. However, this early-semester video access did not translate into sustained practice with the AAT, likely contributing to their lower performance in the final scores. In contrast, both high-performing groups appeared to skip videos initially, beginning directly with practice, and only returned to videos late in the semester (around week 14), while reaching their peak in submissions. A similar but weaker pattern emerged for code examples. Low-performing groups accessed them early in the semester, with an additional peak among low-performing experienced students around weeks 14-16. High-performing groups, by comparison, remained relatively disengaged from code examples, though high-performing novices showed slightly greater engagement during some mid-semester weeks. These patterns reinforce the importance of consistent practice over reliance on videos or code example readings for achieving higher learning outcomes in CS education.

Overall, both high-performing groups began the semester (first 3 weeks) with levels of total activity comparable to their counterparts with similar prior experience (Figure 2D). However, high-performing experienced students were notably less active than low-performing experienced students during the first half of the semester. Around mid-semester, they substantially increased their use of the AAT. From that point on (and especially in the final month), both high-performing groups maintained much higher engagement than their counterparts, averaging over 100 minutes of study per week. High-performing novices reached a peak of roughly 250 minutes per week during the final three weeks. This sustained activity likely contributed to their higher final test scores. In contrast, the other groups showed a steady decline in activity as the semester progressed, suggesting that rising task difficulty, affective states, or other factors may have contributed to their disengagement, leading to lower final test scores.

5 Discussion

This study investigated the interplay between prior programming experience, students' affective states, their engagement with learning resources, and final outcomes in an introductory programming course (CS1). Using structural equation modeling (SEM) and group comparisons, we found that prior knowledge exerted both direct and indirect effects on final test scores. Pre-test and final test scores were strongly correlated, reflecting the known advantage of higher prior expertise in performance during CS1 [41, 42]. However, some novice students who began the semester with significantly less knowledge not only caught up with a group of more experienced peers but even outperformed them on the final tests. This result indicates that although prior programming experience shapes affective experiences and engagement patterns, these factors themselves also contribute to performance in ways that extend beyond the initial advantage of prior knowledge. In particular, persistence in progressing from basic to advanced practice tasks, rather than relying mainly on videos or worked examples, or remaining at introductory tasks, emerged as the strongest predictor of success. This aligns with recent research on online courses proposing that lectures may not be necessary when practice and feedback are available as students who invest time in activities and feedback learn more effectively than those who primarily read or watch videos[2]. This does not imply that lectures or videos should be eliminated; they can still play an important role in scaffolding students until they have the necessary knowledge and skills to learn more independently through practice and feedback. Rather, the main argument is that practice constitutes the most critical component for learning.

5.1 Supporting Novices Through Practice and Scaffolding

Novices' engagement patterns appear to be shaped by the difficulty level of specific tasks. Although confusion was not significantly associated with access to any learning resource, novices reported substantially more confusion than experienced students. The significant correlation between confusion and final test scores (though no longer significant after controlling for prior knowledge) suggests that confusion may be a key factor in their learning. Prior work has noted that confusion does not have a straightforward relationship with learning [15]. For some students, particularly those with the least prior experience, even introductory problems were highly challenging, making it difficult to remain engaged after the first weeks of the semester. By contrast, higher-performing novices, who entered with somewhat stronger prior knowledge, may have drawn on motivation, self-efficacy, or self-regulation to persist practicing despite confusion, though their confusion may have surfaced in more advanced tasks. Although motivation and self-regulation were not measured in this study, future research should investigate whether such factors help explain why some novices continue engaging despite confusion while others disengage.

Another result supporting this interpretation is the role of frustration, often linked to confusion [5, 20] and strongly associated with it in our study. Frustration in novices may have been triggered

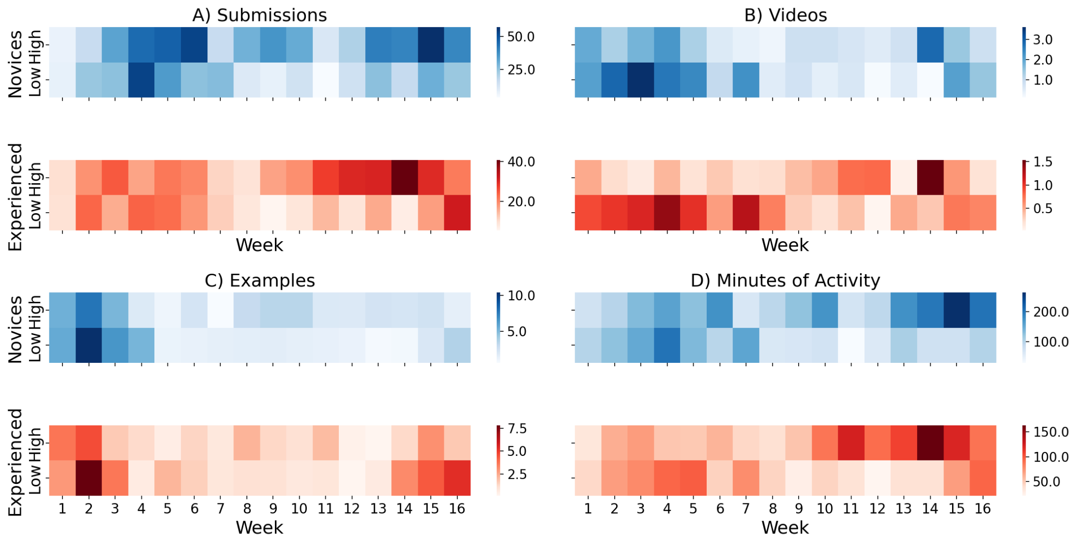

Figure 2: Temporal analysis of access to multiple learning resources over the semester.

by unsuccessful submissions or frequent compiler errors as observed in their activity with the automated assessment tool (AAT). When novices had the motivational resources and self-regulatory skills to manage frustration, they stayed engaged and practiced more—an indirect positive pathway for learning, as indicated in our SEM and echoed in other STEM digital learning environments [23, 43]. However, when frustration did not lead to additional practice, its direct effect was adverse. This dynamic may help explain the generally positive perception of AATs among many students, in contrast to the negative perceptions of some novices who found the feedback to be insufficient [30]. Further analysis of the factors driving self-reports of confusion and frustration, as well as the role of motivation and self-efficacy, is needed to test this hypothesis.

One potential intervention for these students could be to provide additional support, perhaps through Large Language Models (LLMs) that can deliver more targeted and scaffolded feedback when compiler errors or other mistakes occur. Prior research has shown that LLM-based interventions can improve students' correctness in AAT tasks [25] and enhance novices' affective experiences [26]. However, these studies have not yet demonstrated clear benefits on external performance measures and raise concerns about fostering overdependence on such tools. Thus, while LLMs offer promise for provide novices with a more scaffolded learning process that could sustain practice, further research is needed to validate their long-term positive impact on this group of students.

Low-performing novices also tended to access videos and worked examples more than high-performers, which might suggest that videos are not the most effective resource for novices, a pattern also observed among experienced students. However, unlike experienced students who are less likely to encounter confusion, greater video use or reliance on worked examples among novices may instead reflect a need for additional support rather than a genuine preference for these resources or an avoidance of practice, as indicated by the strong positive link between frustration and accessing code examples among these students. For example, Wang et al. [40] found a positive correlation between video activity and coding practice, indicating that novices may primarily use videos to strengthen their understanding of fundamental programming concepts and thereby practice more, rather than less. Further research is needed to clarify whether some novices actively prefer videos and worked examples over practice in AATs, and how instructors or learning systems might better direct students to targeted videos or readings that address common errors or provide the prior knowledge necessary to engage more meaningfully with programming tasks.

5.2 Keeping Experienced Students Challenged and Motivated

Experienced students, in contrast, were less likely to engage consistently with learning resources and reported higher levels of boredom than novices. Within this group, some students remained actively involved (particularly in practice activities) throughout the second half of the semester, achieving the strongest performance as a result. In contrast, those who made little progress tended to disengage after completing only the most basic tasks, without moving on to more challenging exercises. Although many participated actively in the first weeks, they stopped practicing soon after, perhaps because they considered themselves already proficient, a perception that may also explain their boredom [9]. This helps explain the observed sequence in which higher prior knowledge was linked to greater boredom, which in turn reduced practice with AATs and ultimately led to lower final test scores, underscoring the detrimental impact of boredom on learning outcomes. This uniformly negative effect of boredom is consistent with findings from multiple studies (see review in [15]).

Recognizing that this disengagement was likely the result of insufficient challenge in the first weeks of the semester, interventions to foster greater engagement could focus on increasing task difficulty once mastery is evident. This adaptive approach has been widely applied in intelligent tutoring systems, where knowledge tracing is used to present more challenging problems after mastery is detected (e.g., [4, 45]), but it remains underexplored in AATs. Code evaluation adds complexity, since a single task may involve multiple knowledge components, and difficulty may arise from only one of them, making it harder to determine which knowledge components have truly been mastered [35]. Nonetheless, recent research highlights the feasibility of knowledge tracing even in this context [27, 35]. As such, it may be possible to tailor item difficulty to individual students, reducing boredom and improving engagement for more experienced students.

Furthermore, even when some problems were disengaging for experienced students, the fact that others were still able to stay engaged indicates that other motivational factors may buffer against disengagement. For instance, Sun and Rueda [36] and Zhang et al. [44] showed that situational interest is positively correlated with behavioral, cognitive, and emotional engagement. Similarly, Zambrano et al. [43] found that students with high situational interest or self-efficacy were more likely to persist in engaged concentration and less likely to experience boredom. Although these studies did not explicitly control for prior knowledge, it is plausible that some learners who remained highly engaged due to motivational factors already had substantial prior knowledge, pointing to this buffering effect. This hypothesis warrants further investigation and could provide valuable insights for future educational design.

Another factor that may hinder the learning of experienced students is a preference for watching videos, a behavior negatively associated with final test scores, rather than engaging in practice. One possible explanation is that some students in this group have become accustomed to video-based instruction, perhaps through online courses or other contexts where this approach is common. Notably, these students also scored lower on the pre-test. They may believe they have prior knowledge from videos they watched, but in reality, they lack the hands-on practice of their more successful peers. Thus, relying on videos while avoiding practice appears to be a key mechanism limiting their learning. Whether this preference exists and the reasons behind it warrant further investigation to inform potential interventions. If it reflects motivational issues rather than insufficient task difficulty within AATs, interventions should focus on motivation rather than rely solely on knowledge tracing or difficulty adjustments.

6 Conclusion

This study highlights the complex interplay between prior programming experience, affective states, engagement with different learning resources, and performance outcomes in CS1. A consistent finding was the central importance of sustained practice, as students who progressed from basic to advanced tasks achieved the strongest outcomes, whereas reliance on videos or worked examples alone did not predict success and could even hinder learning. Consistent with multiple affective theories [9, 11, 23, 29], novices often struggled with confusion and frustration triggered by compiler errors or excessive task difficulty, while experienced students more often encountered boredom due to insufficient challenge. Although confusion and frustration sometimes led to disengagement, they could also prompt deeper engagement with multiple learning resources. Boredom, by contrast, appeared uniformly detrimental, as experienced students facing it tended to disengage from practice or, in some cases, rely too heavily on videos at the expense of hands-on work.

These patterns underscore the need for interventions that are adaptive with respect to task difficulty and the kind of support offered—for example, scaffolding with worked examples and targeted videos, integrating mechanisms to adjust problem difficulty in AATs, or carefully exploring the potential of LLM-based feedback. Ensuring that all students encounter appropriate levels of challenge, have ample opportunities for hands-on practice, and receive timely support is essential to sustaining engagement. By recognizing and addressing patterns of disengagement, educators can foster more effective learning outcomes in introductory programming courses. Future research should continue to investigate how both cognitive and affective pathways shape learning, with the goal of informing educational designs that support diverse learners in building programming expertise.

Acknowledgments

Andres Felipe Zambrano thanks the Ministerio de Ciencia, Tecnología e Innovación and the Fulbright-Colombia commission for supporting his doctoral studies through the Fulbright-MinCiencias 2022 scholarship.

References

[1] Melanie V. Adler, Jens Madsen, James Hedberg, Richard Steinberg, and Lucas C. Parra. 2025. Effect of explanation videos on learning: The role of attention and academic performance. *Education and Information Technologies* 30, 9 (June 2025), 11797–11825. doi:10.1007/s10639-024-13292-9

[2] Michael W Asher, Faria Sana, Kenneth R Koedinger, and Paulo F Carvalho. 2025. Practice with feedback versus lecture: Consequences for learning, efficiency, and motivation. *Journal of Applied Research in Memory and Cognition* (2025). Publisher: Educational Publishing Foundation.

[3] Burcak Aydin, Gökhan Akcapinar, Vildan Özeke, and Mohammad Nehal Hasnine. 2024. Dynamics of Students' Affective States and Video Interactions While Watching Educational Videos. *International Association for Development of the Information Society* (2024). Publisher: ERIC.

[4] Ryan Baker, Wei Ma, Yuxin Zhao, Shengni Wang, and Zhenjun Ma. 2020. The Results of Implementing Zone of Proximal Development on Learning Outcomes. *International Educational Data Mining Society* (2020). Publisher: ERIC.

[5] Ryan S. Baker, Elizabeth Cloude, Juliana M. A. L. Andres, and Zhanlan Wei. 2025. The Confrustion Constellation: A New Way of Looking at Confusion and Frustration. *Cognitive Science* 49, 1 (Jan. 2025), e70035. doi:10.1111/cogs.70035 Publisher: John Wiley & Sons, Ltd.

[6] Nigel Bosch and Sidney D'Mello. 2017. The Affective Experience of Novice Computer Programmers. *International Journal of Artificial Intelligence in Education* 27, 1 (March 2017), 181–206. doi:10.1007/s40593-015-0069-5

[7] Giang Bui, Naaz Sibia, Angela Zavaleta Bernuy, Michael Liut, and Andrew Petersen. 2023. Prior Programming Experience: A Persistent Performance Gap in CS1 and CS2. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 889–895. doi:10.1145/3545945.3569752 event-place: Toronto ON, Canada.

[8] Sébastien Combéfis. 2022. Automated Code Assessment for Education: Review, Classification and Perspectives on Techniques and Tools. *Software* 1, 1 (2022), 3–30. doi:10.3390/software1010002

[9] Mihaly Csikszentmihalyi. 1990. *Flow: The psychology of optimal experience*. Vol. 1990. Harper & Row New York.

[10] Ivana Di Leo, Krista R Muis, Cara A Singh, and Cynthia Psaradellis. 2019. Curiosity… Confusion? Frustration! The role and sequencing of emotions during mathematics problem solving. *Contemporary educational psychology* 58 (2019), 121–137. Publisher: Elsevier.

[11] Sidney D'Mello and Art Graesser. 2012. Dynamics of affective states during complex learning. *Learning and Instruction* 22, 2 (2012), 145–157. doi:10.1016/j.learninstruc.2011.10.001

[12] Jennifer A Fredricks, Phyllis C Blumenfeld, and Alison H Paris. 2004. School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research* 74, 1 (March 2004), 59–109. doi:10.3102/00346543074001059 Publisher: American Educational Research Association.

[13] Dianne Hagan and Selby Markham. 2000. Does it help to have some programming experience before beginning a computing degree program?. In *Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSEconference on Innovation and Technology in Computer Science Education (ITiCSE '00)*. Association for Computing Machinery, New York, NY, USA, 25–28. doi:10.1145/343048.343063 event-place: Helsinki, Finland.

[14] Li-tze Hu and Peter M. Bentler. 1999. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal* 6, 1 (Jan. 1999), 1–55. doi:10.1080/10705519909540118 Publisher: Routledge.

[15] Shamya Karumbaiah, Ryan Baker, Yan Tao, and Ziyang Liu. 2022. How does Students' Affect in Virtual Learning Relate to Their Outcomes ? A Systematic Review Challenging the Positive-Negative Dichotomy. In *LAK22: 12th International Learning Analytics and Knowledge Conference (LAK22)*. Association for Computing Machinery, New York, NY, USA, 24–33. doi:10.1145/3506860.3506863

[16] Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. 2018. A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Trans. Comput. Educ.* 19, 1 (Sept. 2018). doi:10.1145/3231711 Place: New York, NY, USA Publisher: Association for Computing Machinery.

[17] Shelbi L. Kuhlmann, Robert Plumley, Zoe Evans, Matthew L. Bernacki, Jeffrey A. Greene, Kelly A. Hogan, Michael Berro, Kathleen Gates, and Abigail Panter. 2024. Students' active cognitive engagement with instructional videos predicts STEM learning. *Computers & Education* 216 (July 2024), 105050. doi:10.1016/j.compedu.2024.105050

[18] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. 2005. A study of the difficulties of novice programmers. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '05)*. Association for Computing Machinery, New York, NY, USA, 14–18. doi:10.1145/1067445.1067453 event-place: Caparica, Portugal.

[19] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. John Wiley & Sons.

[20] Zhongxiu Liu, Visit Pataranutaporn, Jaclyn Ocumpaugh, and Ryan Baker. 2013. Sequences of frustration and confusion, and learning. In *International Conference on Educational Data Mining*. 114–120.

[21] Zachary MacHardy and Zachary A Pardos. 2015. Evaluating the Relevance of Educational Videos Using BKT and Big Data. *International Educational Data Mining Society* (2015). Publisher: ERIC.

[22] Michael Noetel, Shantell Griffith, Oscar Delaney, Taren Sanders, Philip Parker, Borja del Pozo Cruz, and Chris Lonsdale. 2021. Video Improves Learning in Higher Education: A Systematic Review. *Review of Educational Research* 91, 2 (April 2021), 204–236. doi:10.3102/0034654321990713 Publisher: American Educational Research Association.

[23] Jaclyn Ocumpaugh, Nidhi Nasiar, Andres Felipe Zambrano, Alex Goslen, Jessica Vandenberg, Jordan Esiason, Jonathan Rowe, and Stepehn Hutt. 2025. Refocusing the lens through which we view affect dynamics: The Skills, Difficulty, Value, Efficacy and Time Model. Dublin, Ireland.

[24] Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013*. ACM/Association for Computing Machinery.

[25] Maciej Pankiewicz and Ryan S Baker. 2023. Large Language Models (GPT) for automating feedback on programming assignments. In *International Conference on Computers in Education*.

[26] Maciej Pankiewicz and Ryan S. Baker. 2024. Navigating Compiler Errors with AI Assistance - A Study of GPT Hints in an Introductory Programming Course. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 94–100. doi:10.1145/3649217.3653608 event-place: Milan, Italy.

[27] Maciej Pankiewicz, Yang Shi, and Ryan S. Baker. 2025. srcML-DKT: Enhancing Deep Knowledge Tracing with Robust Code Representations from srcML. In *Proceedings of the 18th International Conference on Educational Data Mining*, Caitlin Mills, Giora Alexandron, Davide Taibi, Giosuè Lo Bosco, and Luc Paquette (Eds.). International Educational Data Mining Society, Palermo, Italy, 541–548. doi:10.5281/zenodo.15870306

[28] Maciej Pankiewicz, Andres Felipe Zambrano, Amanda Barany, and Ryan S. Baker. 2024. How We Code Code: Leveraging GPT and Ordered Networks for Understanding Introductory Programming Education. In *Advances in Quantitative Ethnography*, Yoon Jeon Kim and Zachari Swiecki (Eds.). Springer Nature Switzerland, Cham, 225–240.

[29] Reinhard Pekrun. 2006. The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. *Educational Psychology Review* 18, 4 (Dec. 2006), 315–341. doi:10.1007/s10648-006-9029-9

[30] Raymond Scott Pettit, John D Homer, Kayla Michelle McMurry, Nevan Simone, and Susan A Mengel. 2015. Are automated assessment tools helpful in programming courses?. In *2015 ASEE Annual Conference & Exposition*. 26–230.

[31] Sanjay Rajesh, Vineeth V Rao, and MG Thushara. 2024. Comprehensive Investigation of Code Assessment Tools in Programming Courses. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*. 1–6. doi:10.1109/I2CT61223.2024.10543863

[32] Ma. Mercedes T. Rodrigo, Ryan S. Baker, Matthew C. Jadud, Anna Christine M. Amarra, Thomas Dy, Maria Beatriz V. Espejo-Lahoz, Sheryl Ann L. Lim, Sheila A.M.S. Pascua, Jessica O. Sugay, and Emily S. Tabanao. 2009. Affective and behavioral predictors of novice programmer achievement. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*. Association for Computing Machinery, New York, NY, USA, 156–160. doi:10.1145/1562877.1562929 event-place: Paris, France.

[33] Adrian Salguero, Julian McAuley, Beth Simon, and Leo Porter. 2020. A Longitudinal Evaluation of a Best Practices CS1. In *Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER '20)*. Association for Computing Machinery, New York, NY, USA, 182–193. doi:10.1145/3372782.3406274 event-place: Virtual Event, New Zealand.

[34] Sreecharan Sankaranarayanan, Siddharth Reddy Kandimalla, Christopher Bogart, R Charles Murray, Haokang An, Michael Hilton, Majd Sakr, and Carolyn Rosé. 2021. Comparing Example-Based Collaborative Reflection to Problem-Solving Practice for Learning during Team-Based Software Engineering Projects. In *Proceedings of the 14th International Conference on Computer-Supported Collaborative Learning-CSCL 2021*. International Society of the Learning Sciences.

[35] Yang Shi, Min Chi, Tiffany Barnes, and Thomas Price. 2022. Code-DKT: A Code-based Knowledge Tracing Model for Programming Tasks. In *Proceedings of the 15th International Conference on Educational Data Mining*, Antonija Mitrovic and Nigel Bosch (Eds.). International Educational Data Mining Society, Durham, United Kingdom, 50–61. doi:10.5281/zenodo.6853105

[36] Jerry Chih-Yuan Sun and Robert Rueda. 2012. Situational interest, computer self-efficacy and self-regulation: Their impact on student engagement in distance education. *British Journal of Educational Technology* 43, 2 (March 2012), 191–204. doi:10.1111/j.1467-8535.2010.01157.x Publisher: John Wiley & Sons, Ltd.

[37] Massimiliano Tani, Maurizio Manuguerra, and Samia Khan. 2022. Can videos affect learning outcomes? Evidence from an actual learning environment. *Educational technology research and development* 70, 5 (Oct. 2022), 1675–1693. doi:10.1007/s11423-022-10147-3

[38] Ashok Kumar Veerasamy, Daryl D'Souza, Rolf Lindén, and Mikko-Jussi Laakso. 2018. The Impact of Prior Programming Knowledge on Lecture Attendance and Final Exam. *Journal of Educational Computing Research* 56, 2 (April 2018), 226–253. doi:10.1177/0735633117707695 Publisher: SAGE Publications Inc.

[39] Haipeng Wan, Xue Zhang, Xinxue Yang, and Shan Li. 2024. Which approach is effective: Comparing problematization-oriented and structuring-oriented scaffolding in instructional videos for programming education. *Education and Information Technologies* 29, 14 (Oct. 2024), 17807–17823. doi:10.1007/s10639-024-12550-0

[40] Xiaonan Wang, Yancong Su, Yi Sun, Kento Shigyo, Kazuhiro Ohtsuki, and Hidenari Kiyomitsu. 2024. Navigating Learning Pathways: A Multifaceted Approach to Understanding Student Engagement and Progress in Online Programming Courses. In *2024 12th International Conference on Information and Education Technology (ICIET)*. 501–506. doi:10.1109/ICIET60671.2024.10542762

[41] Chris Wilcox and Albert Lionelle. 2018. Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 80–85. doi:10.1145/3159450.3159480 event-place: Baltimore, Maryland, USA.

[42] Brenda Cantwell Wilson and Sharon Shrock. 2001. Contributing to success in an introductory computer science course: a study of twelve factors. In *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education (SIGCSE '01)*. Association for Computing Machinery, New York, NY, USA, 184–188. doi:10.1145/364447.364581 event-place: Charlotte, North Carolina, USA.

[43] Andres Felipe Zambrano, Jaclyn Ocumpaugh, Ryan S. Baker, Kirk Vanacore, Jordan Esiason, and Jessica Vandenberg. 2025. The Half-Life of Epistemic Emotions: How Motivation Influences Affective Chronometry. In *Proceedings of the 18th International Conference on Educational Data Mining*. International Educational Data Mining Society, Palermo, Italy.

[44] Jiayi Zhang, Stephen Hutt, Jaclyn Ocumpaugh, Nathan Henderson, Alex Goslen, Jonathan P. Rowe, Kristy Elizabeth Boyer, Eric Wiebe, Bradford Mott, and James Lester. 2022. Investigating Student Interest and Engagement in Game-Based Learning Environments. In *Artificial Intelligence in Education*, Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova (Eds.). Springer International Publishing, Cham, 711–716.

[45] Xiaotian Zou, Wei Ma, Zhenjun Ma, and Ryan S. Baker. 2019. Towards Helping Teachers Select Optimal Content for Students. In *Artificial Intelligence in Education*, Seiji Isotani, Eva Millán, Amy Ogan, Peter Hastings, Bruce McLaren, and Rose Luckin (Eds.). Springer International Publishing, Cham, 413–417.