2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #52625

Analysis of Student Engagement and Behavior Across A Spectrum of Interactive Learning Activities in an Introductory Programming Course.

Jamie Emily Loeber, zyBooks, A Wiley Brand

Jamie Loeber is an Assessment Specialist at zyBooks, a Wiley Brand. She earned her B.S. in Computer Science at the University of California, Irvine. She has taught programming and machine learning to students across the globe. Jamie is passionate about improving computer science education and creating better learning experiences in STEM.

Dr. Annie Wood, zyBooks, A Wiley Brand

Annie Hui is a zyBooks assessment specialist. She has 15 years of experience teaching computer science, information technology, and data science courses, in both in-person and online modes. She has taught in Northern Virginia Community College and George Mason University. She specializes on course design to maximize student engagement and success.

Nicole Kehaulani Collins, zyBooks, A Wiley Brand

Nicole Collins is an Author Trainer and former Assessment Specialist at zyBooks, a Wiley Brand. She earned her B.S. in Computer Science and her M.Ed. in Learning, Design & Technology from UNC Charlotte. Her professional interests include computing education, online learning, educational technology, instructional design, curriculum development, and DEI in STEM. Nicole is passionate about creating engaging and effective learning experiences for students, leveraging her expertise in instructional design and technology to enhance educational outcomes for STEM disciplines.

Dr. Yamuna Rajasekhar, zyBooks, A Wiley Brand

Yamuna Rajasekhar is a senior manager of Content at zyBooks, a Wiley Brand. She is an author and contributor to various zyBooks titles. She was formerly an assistant professor of Electrical and Computer Engineering at Miami University. She received her M.S. and Ph.D. in Electrical and Computer Engineering from UNC Charlotte.

Analysis of Student Engagement and Behavior Across A Spectrum of Interactive Learning Activities in an Introductory Programming Course

Abstract

Over the past decade, computer science education has undergone a significant digital transformation, with interactive activities becoming a core feature of many introductory programming textbooks, designed to help students actively engage with new concepts. These interactive activities can accommodate various learning styles through differentiated instruction such as visualizations of complex algorithms, formative assessments, and advanced coding practice problems, all of which reinforce learning through reasoning. However, increased cognitive load can impact learning negatively. The optimal amount and types of interactive activities that best support learning without any negative impacts are not well established. Finding the right balance is especially important for introductory students, who are often learning to program for the first time.

This paper examines patterns of student engagement and behavior to investigate the optimal level of interactivity while maintaining a positive impact on student learning. We identify which design choices encourage active participation with a positive effect on student performance, and which may create barriers for newer students.

We focus on a widely used introductory Python textbook and select several high-enrollment sections that feature varying degrees of interactive activities, such as animations, along with formative assessments and autograded coding activities that provide immediate and meaningful feedback. Using the activity data, we analyze metrics including students' time spent, number of attempts, and completion rates to understand how students interact with sections that differ in structure and intensity. We also study the interaction patterns of students who use accessibility features in the same introductory Python textbook to understand how accessibility features impact student performance.

Interactive activities can improve engagement, but excessive interactivity may increase cognitive load and reduce effectiveness, thereby negatively influencing student behavior. By identifying engagement patterns across thousands of students, this work aims to provide practical insights into how much interactivity is most effective, guiding future digital textbook and course material design. This work offers insights on student behavior with accessible content and guidelines to create more inclusive digital material.

Background and Motivation

Interactive Activities in Programming Education

In the computer science discipline, digital programming platforms are prevalent. These platforms incorporate interactive activities such as visualizations, formative assessments, and autograded coding exercises with explanatory text. Unlike traditional static textbooks, these platforms allow students to directly manipulate and run code within the learning environment. The design of these platforms draws from constructivist learning theories, which propose that students construct knowledge through active engagement with material [1, 2]. Program visualizations have been studied as tools for helping students understand code execution [3], and automated feedback systems provide immediate responses to student submissions [4].

Cognitive Load Considerations

Cognitive Load Theory proposes that working memory capacity is limited and that instructional design should account for these limitations [5, 6]. In interactive programming materials, interactive activities might help clarify abstract concepts, but they could also add unnecessary complexity if students have a steep learning curve for a new tool while simultaneously learning new programming concepts. Novice programmers already face the challenge of learning syntax, semantics, problem-solving strategies, and development tools all at once. Whether interactive features support or complicate the learning process likely depends on both design choices and individual student characteristics.

Introductory Programming Context

Introductory programming (CS1) courses have traditionally documented high failure and withdrawal rates, with studies across multiple institutions reporting rates ranging from 30-50% [7, 8]. Many factors influence these outcomes, including prior preparation, pedagogy, course structure, and instructional materials. Students enter CS1 with varying levels of prior experience, from none to substantial. This broad range means that learning materials may affect different students in quite different ways.

Accessibility Considerations

Students using accessibility features such as screen readers or keyboard navigation interact with digital content differently than students who do not use such features. Digital platforms can provide accommodations and support students in ways not possible in print textbooks.

Study Focus

This study examines student engagement patterns in a digital introductory Python textbook that includes various interactive activities, analyzing metrics such as time spent, attempts, and completion rates across sections with different levels and types of interactivity. We also examine engagement patterns among students using accessibility features in all subjects across the platform. Specifically, this study addresses the following questions:

1. How does the presence of animations in textbook sections affect student engagement and time spent on assessments?
2. How do different types of assessment activities differ in terms of student engagement metrics including time spent and attempts required?
3. How does accessibility feature usage vary across different course contexts such as class size, academic level, and subject area?
4. How do accessibility users differ from non-accessibility users in terms of engagement and performance outcomes?

From these findings, we offer insights for optimizing interactive design to improve learning outcomes.

Methodology

In our study on the impact of interactivity design on learning behavior, we examined interactivity design in a single high-enrollment introductory Python textbook. We focused on a single textbook to reduce variation in content style, assessment design, and course structure. This allowed us to study how different combinations of interactive activities relate to student engagement and performance.

The introductory Python textbook included 163 sections spanning a range of interactive densities, from 0-20 activities per section. The dataset included activity-level and section-level data collected between the Spring 2024 and Fall 2025 academic terms, covering 6,013 students and their interactions across multiple courses.

The analysis only included sections that contained interactive activities and excluded sections with no recorded activity data such as time spent and activity attempts.

Interactive Activities and Section Structure

Each section contained a combination of interactive activities, including challenge activities (CAs), animations, and formative assessments.

Challenge activities (CAs):

Four types of CAs were studied:

- *CodeOutput CAs* assess a student's ability to read and analyze provided code to predict the output.

- *CodeWriting CAs* assess a student's ability to write short code snippets to complete a program.
- *CodeWarmup CAs* are similar to CodeWriting CAs but are designed to introduce students to simple problems before progressing to the more complex ones.
- *ProblemSolving CAs* encompass a wide collection of conceptual, analytical, and logical reasoning problems presented in diverse formats including multiple choice, select-all-that-apply, and fill-in-the-blank.

CodeOutput, CodeWriting, and ProblemSolving CAs are highly randomized so that students can practice repeatedly to achieve mastery. All CAs are autograded for completion, and each attempt of a CA is recorded, as well as how many students were able to successfully complete the CA.

Animations:

Animations provide visual representations of programming operations for instruction purposes. Animations were treated as a distinct category due to their non-assessment nature as they are specifically designed to engage students by starting them and navigating through each step. We counted watching an animation as student engagement.

Formative assessments:

Formative assessments are questions embedded within sections for instruction purposes. Such questions were included as interactive activities at the section level but excluded from attempt-based performance metrics, since they do not follow the same correctness and retry structure as CAs. When a student selects an incorrect answer, the formative assessment advises them as to why their selection was incorrect and allows students unlimited attempts to select the correct answer. These questions are not required for students.

Measures of Interactivity

Section interactivity was measured by two parameters:

- ***Total interactivity count:*** The sum of unique CAs and formative assessment questions in a section.
- ***Animation presence:*** The presence of at least one animation within a section.

Engagement and Performance Metrics

Student engagement and performance were measured by the following metrics:

- ***Attempts:*** A student submission to a CA that is evaluated for correctness by the platform. For sections composed of multiple CAs, attempts were aggregated to the section level.
- ***Number of attempts until correct:*** The number of attempts a student needs to reach a correct solution, calculated for each CA, then aggregated to the section level.
- ***Time spent:*** Total elapsed time a student spent interacting with all CAs within a section, measured from first interaction to final submission, aggregated at the section level.

Time-based analyses were trimmed at the 99th percentile per CA to reduce the influence of outliers.

Sections were grouped according to the number of interactive activities they contained. Across the textbook, sections ranged from 0 to 20 interactive activities, with the majority clustering between 2 and 5 activities and only a small number of sections exceeding 15 activities. To capture differences in engagement and performance while avoiding sparsely populated groups, sections were binned into four interactivity ranges: 1-2, 3-4, 5-7, and 8+ interactive activities. These groupings ensured adequate sample sizes while enabling comparisons across the interactivity spectrum. Student engagement and performance metrics were then compared across bins.

Accessibility

Engagement on different sections throughout the introductory Python textbook were analyzed based on whether students have accessibility features enabled. Students who voluntarily enabled accessibility features form the group of accessibility users, while those who opted out of the features form the group of non-accessibility users. Of the 6,013 students across 163 sections, 712 (11.84%) had accessibility features enabled. For each of the four CA types (CodeOutput, CodeWriting, CodeWarmup, and ProblemSolving), student engagement was measured using time spent and number of attempts until correct. To assess how design choices affected engagement, sections were categorized by animation presence, and average time spent per CA was compared across for both accessibility and non-accessibility users. Differences in number of attempts until correct were evaluated using a Mann–Whitney U test due to skewed distributions, with effect sizes reported using Cohen's d and rank-biserial correlation.

To better understand accessibility feature usage, we conducted a platform-wide analysis extending beyond the introductory Python textbook. This analysis included 13,136 courses with 9,281 accessibility users and 81,079 non-accessibility users. This platform-wide analysis examined the following six aspects of accessibility usage patterns.

Year-to-year trends:

Accessibility usage was compared between 2024 and 2025 to identify usage change platform-wide.

Class size impact:

Accessibility ratio was defined as the proportion of accessibility users within each class. Courses were categorized into six class size brackets: very small (1-5), small (6-50), medium (51-100), large (101-150), very large (151-200), and extra large (200+). For each bracket, the mean, median, standard deviation of accessibility ratio were calculated.

Course level impact:
Accessibility usage was analyzed across five course levels. Levels 1 through 4 correspond to the four typical years of undergraduate studies. The Graduate/Other Level includes all graduate studies and courses whose level cannot be determined based on course code. The accessibility user count and percentage were calculated for each level.

Subject-based impact:
Courses were categorized into 10 major subject areas:

1. Programming and Software Development
2. Computer Science Fundamentals
3. Algorithms, Data Structures, and Discrete Math
4. Engineering Sciences (including material science and engineering foundations)
5. Computer Systems and Architecture
6. Data Science and Analytics
7. Mathematics, Statistics, and Scientific Computing
8. Web and Mobile Development
9. Electrical and Computer Engineering
10. Information Technology and Security

Courses not in a major subject area were classified as "Other". For each subject, we examined accessibility usage patterns in both very small classes with at most 5 students and regular classes with over 5 students.

Student engagement through feedback:
A platform-wide comparison was made between accessibility users and non-accessibility users in terms of positive and negative section feedback and bug reporting rates, taking into account that some users in both groups contributed to one or more categories of feedback and bug reporting while others none at all.

Overall performance assessment:
Platform-wide, 9,410 accessibility users and 72,545 non-accessibility users completed CAs. Some students appeared in both groups as they used accessibility features in some but not all enrolled classes. Students who completed CAs were categorized into four performance categories based on their average CA scores: Struggling <70%, Developing 70-85%, Good 85-95%, Excellent ≥95%. The distribution differences of the two groups across these categories was analyzed using a chi-square test of independence with Cramér's V for effect size.

<u>Limitations</u>
This study is observational and does not control for instructor practices, grading policies, or student backgrounds. The analysis focuses on behavioral patterns and design tradeoffs rather

than direct learning outcomes. We did not adjust for section length or content difficulty, noting that interactivity correlates with each.

Results

<u>Interactivity</u>

Sections varied widely in total interactivity, defined as the sum of CAs, animations, and formative questions. Across the 163 sections analyzed, total interaction ranged from 0 to 20 activities, with most sections clustered between 2 and 7 activities (**Figure 1**). Because sections with very high interactivity were less common and the distribution is right skewed, we grouped sections into four bins (1-2, 3-4, 5-7, 8+) to stable sample sizes in each group (**Table 1**). These bins separate low, moderate, and high interactivity while avoiding unreliable estimates in the sparse upper range.

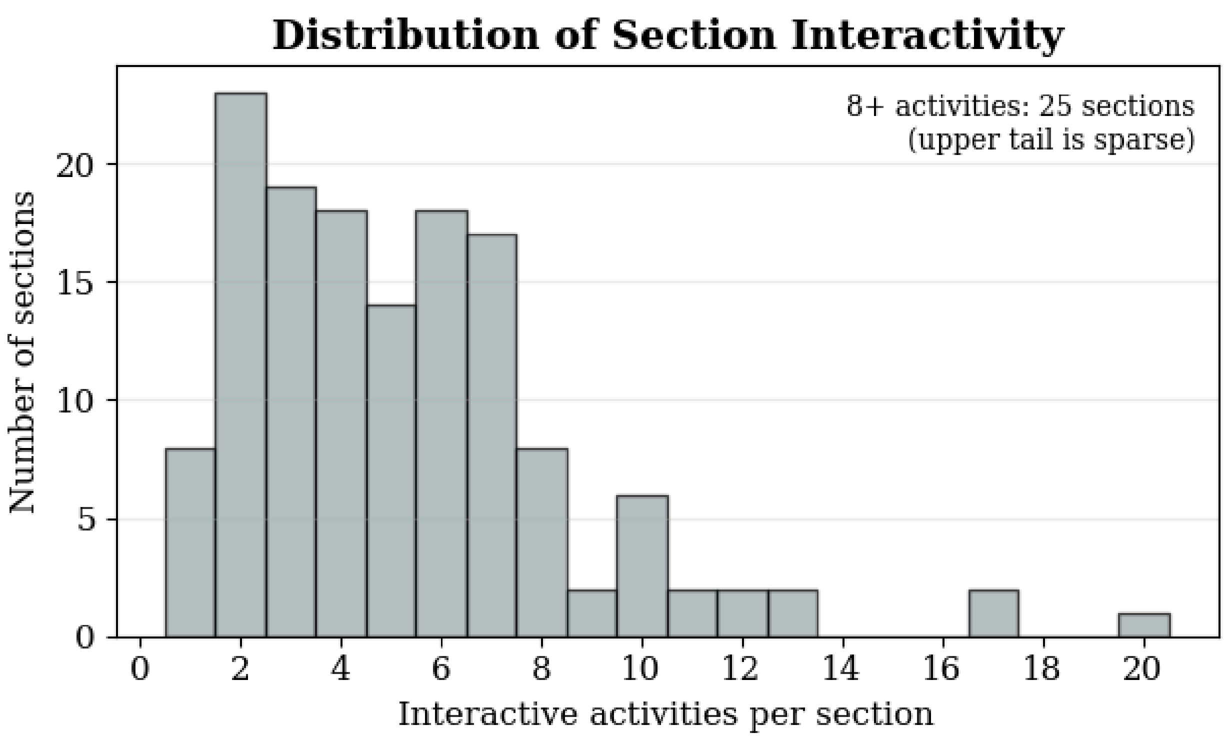

Figure 1. Distribution of interactive activities across 20 sections.

Table 1. Binned distribution of interactive activities per section.

Interactivity bin (activities per section)	Number of Sections (n)	Percent of Sections
1-2	31	21.8%
3-4	37	26.1%
5-7	49	34.5%
8+	25	17.6%

Time spent per section increased sharply as interactivity increased (**Figure 2**). This increase is driven in part by the small number of sections in the upper range, which include some sections with very high activity counts (17-20 activities). To reduce bias to those potential outliers, we report aggregated results for the 8+ group. The pattern indicates that sections with higher interactivity are associated with greater time spent overall, reflecting engagement and sustained effort, though higher interaction complexity may also contribute.

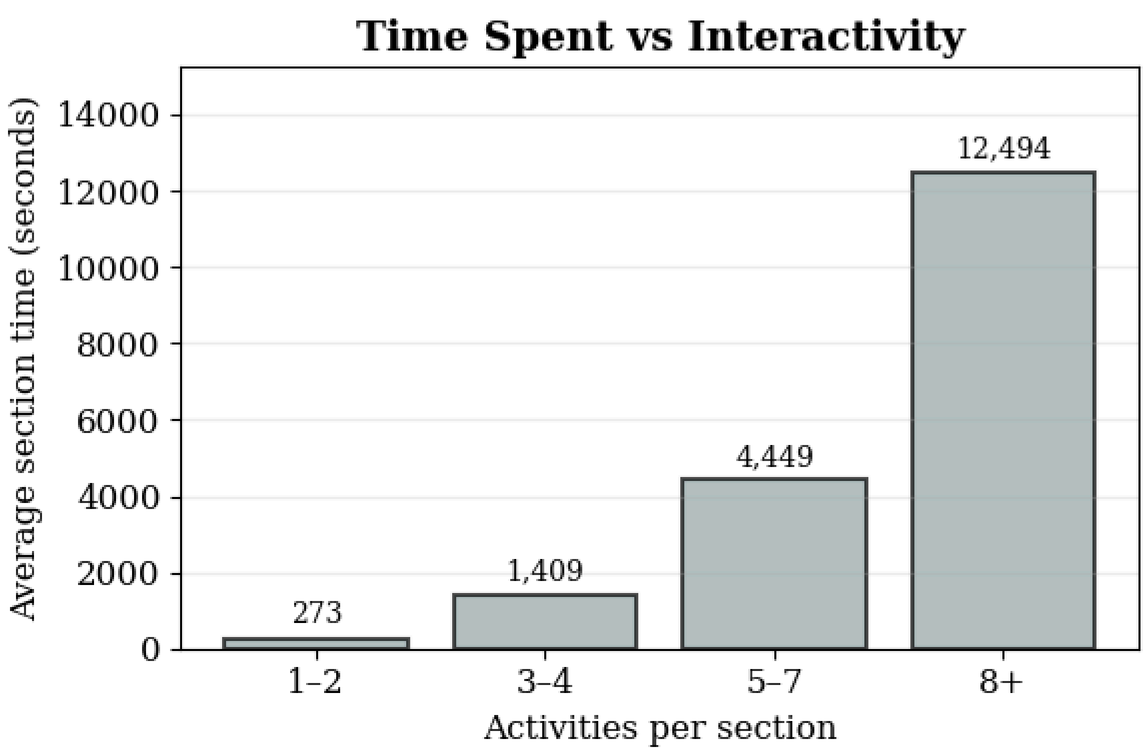

Figure 2. Average time spent per section by interactivity level.

Average attempts to reach a correct solution increases as section interactivity rises, with sections containing 1-2 activities showing the lowest mean attempts and sections with 8 or more activities showing the highest (**Figure 3**). The pattern across each bin reflects a gradual increase in persistence or effort rather than abrupt changes in task difficulty. However, 95% confidence intervals overlap across adjacent interactivity levels, suggesting that differences between neighboring bins are small. Overall, higher interactivity is associated with slightly greater engagement, reflected in increased attempts, without evidence of substantial performance decline.

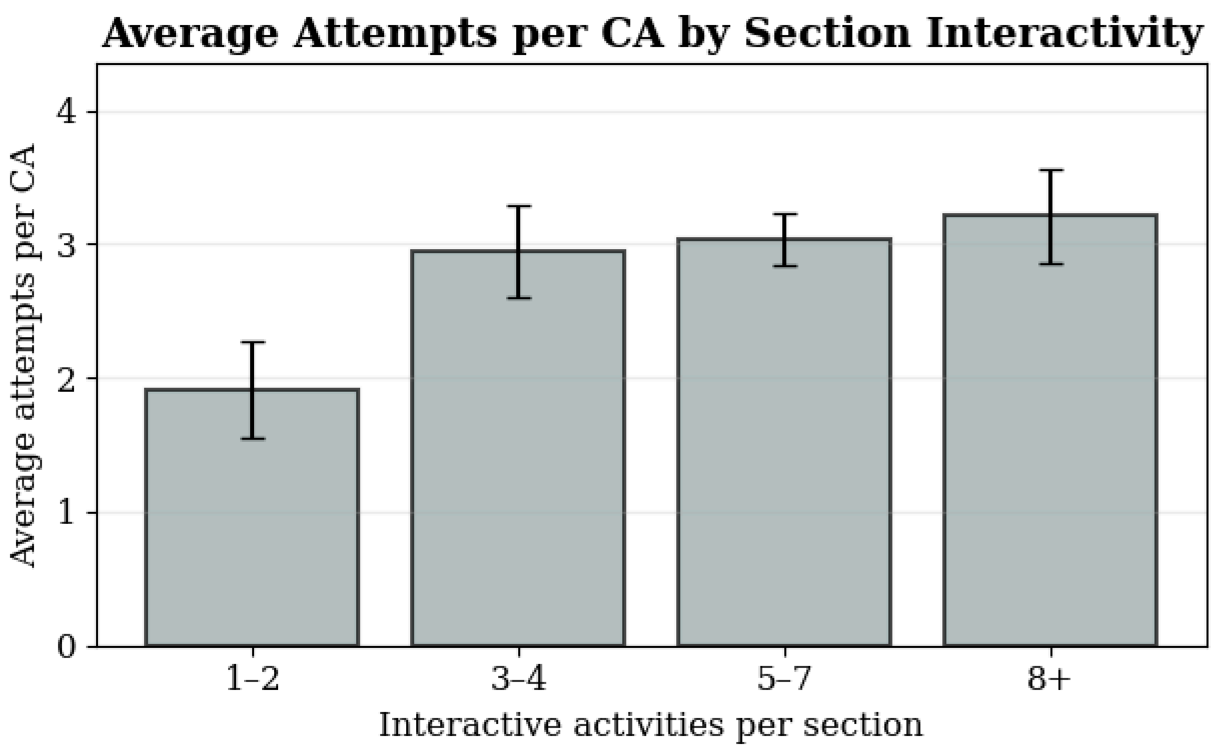

Figure 3. Average attempts per Challenge Activity by section interactivity level. Error bars indicate 95% confidence intervals.

Average attempts vary across levels of section interactivity, with no clear rise or drop across the full range (**Figure 4).** Low and moderate interactivity sections show a wide spread in average attempts, while higher interactivity sections show similar values rather than consistently higher ones. The linear trend shows only a small positive slope, suggesting that adding more interactive activities is not strongly linked to changes in how many attempts students make.

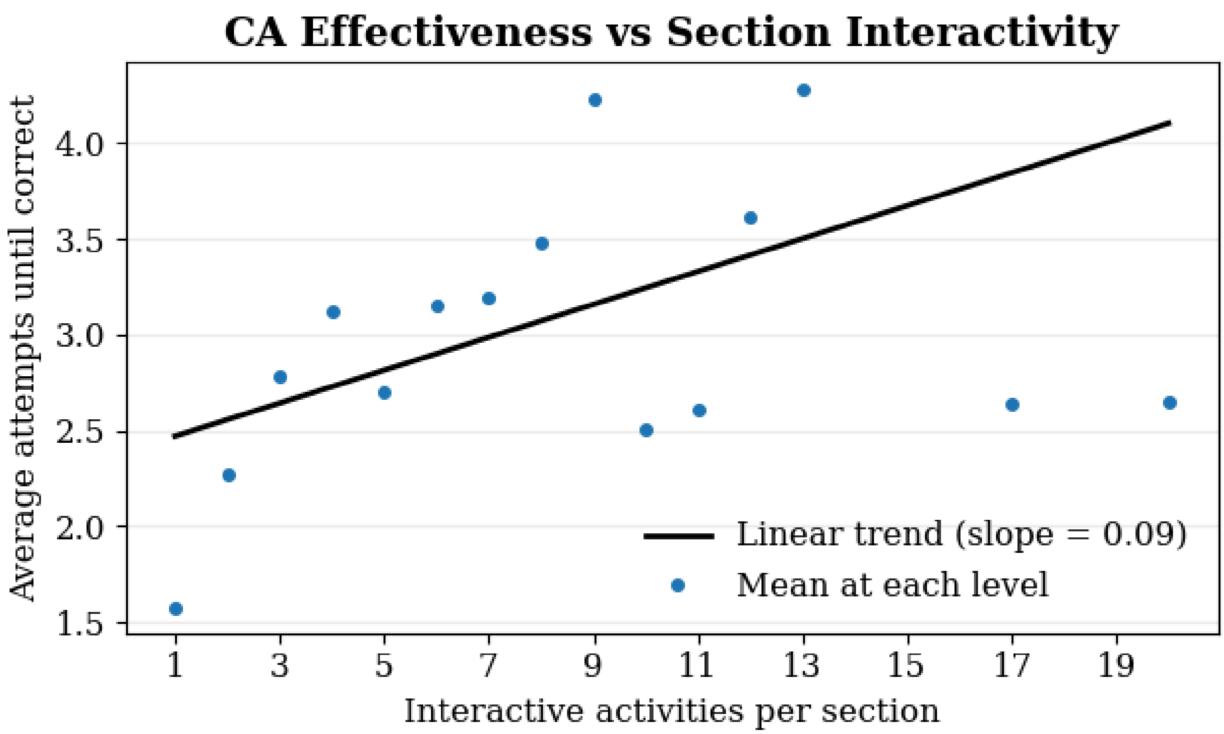

Figure 4. Average attempts until correct per Challenge Activity by section interactivity level, with linear trend line.

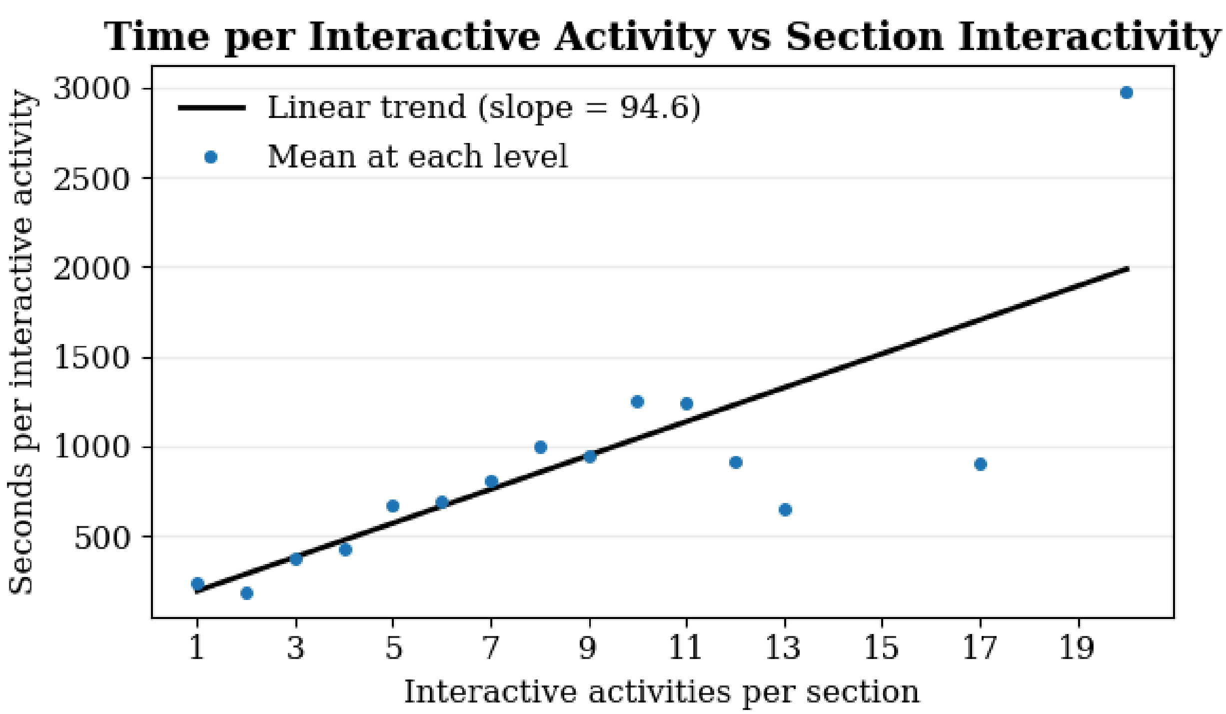

Figure 5. Average time per interactive activity by section interactivity level, with linear trend line.

In contrast to the decline in attempts per activity, time spent per interactive activity increased with section interactivity (**Figure 5**). From low to medium interactivity levels, each activity required more time on average, suggesting longer engagement or planning between submissions. At higher interactivity levels, the trend becomes less stable, reflecting the sparse number of high interactive sections.

Together, **Figure 4** and **Figure 5** show that the average number of attempts until correct remains near 3 attempts across most interactivity levels, while the time spent per activity steadily increases at a rate of about 94.6 (or approximately 100) seconds per activity for sections with 2 to 9 activities. This indicates that, within this range, greater time spent does not correspond to a significant change in the number of attempts needed to pass. Beyond approximately 9 activities, both time and performance become more variable, limiting interpretability in the upper range.

Accessibility

The following are the results of the study on student engagement with accessibility features within the context of the introductory Python textbook.

Accessibility-enabled students spent more time on CAs than non-accessibility users, averaging 1,484 seconds per CA, compared to 1,053 seconds for non-accessibility users (**Figure 6(a)**), indicating higher sustained effort when working through interactive content. Accessibility users also required slightly more attempts to reach a correct solution. On average, accessibility users required 3.03 attempts, compared to 2.91 attempts for non-accessibility users. (**Figure 6(b)**). This difference was statistically significant according to a Mann-Whitney test with p-value of 0.0015, but the magnitude of the difference was very small, corresponding to an average increase of only 0.12 attempts. Effect sizes were negligible, with Cohen's $d = 0.04$ and a rand-biserial $r = 0.009$, indicating that accessibility users persist slightly longer without experiencing meaningfully worse outcomes.

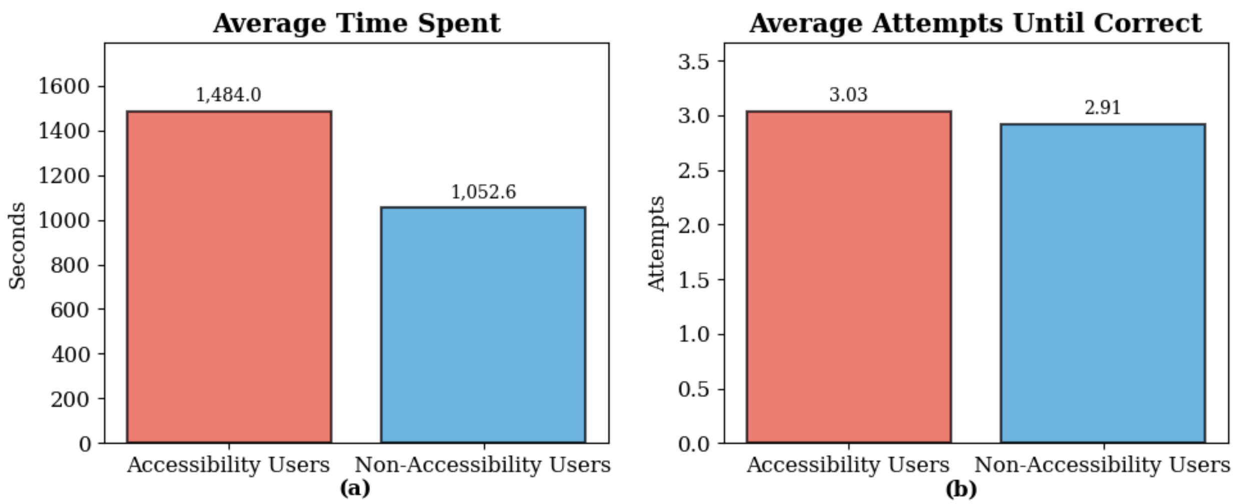

Figure 6. Comparison of accessibility and non-accessibility users: (a) average time spent per section in seconds, (b) average attempts until correct per Challenge Activity.

When attempts are aggregated at the CA level, accessibility users make more attempts per CA on average than non-accessibility users, with mean of 9.2 and 7.4 attempts respectively. This difference is statistically significant according to a Mann-Whitney U test with a p-value below 0.0049, and a modest rank-biserial correlation of -0.21. However, typical behavior is similar across groups. As shown in the boxplot (**Figure 7**) median attempts and interquartile ranges largely overlap, with differences concentrated in the upper tail of the distribution. The higher mean for accessibility users is driven by increased persistence in the upper tail of the distribution rather than by widespread difficulty. Overall, these results indicate that accessibility users persist longer on CAs without evidence of worse performance.

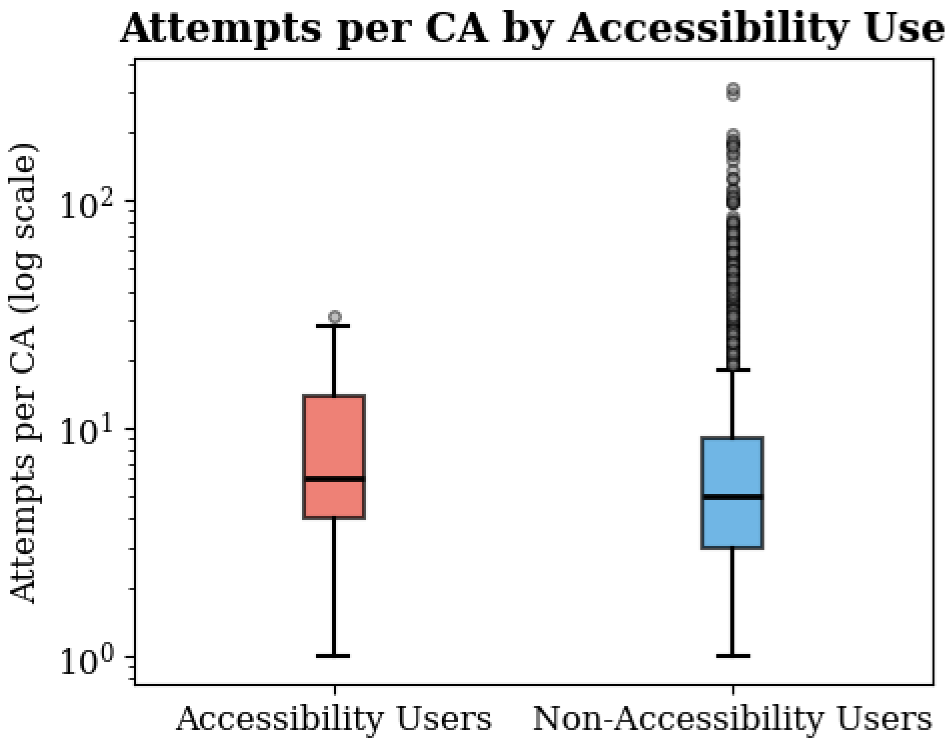

Figure 7. Box plot of attempts per Challenge Activity for accessibility and non-accessibility users (log scale).

Differences in average time spent per CA vary by animation presence and accessibility use (**Figure 8**). Accessibility users spend more time than non-accessibility users in both conditions, with an average difference of approximately 70 seconds for sections without animation (285 sec vs. 214 sec) and 30 seconds for sections with animation (108 sec vs 76 sec). Sections with no animation show a wider spread of CA time, this could be due to less understanding of the material, or more text in place of the lack of animation. Across both groups, sections with animations are associated with lower average CA time, indicating faster completion. The distributional view further shows that animation presence is associated with lower median CA time, particularly for accessibility users. (**Figure 8**) Together, these results suggest that animations reduce overall CA time spent and partially narrow accessibility-related effort differences.

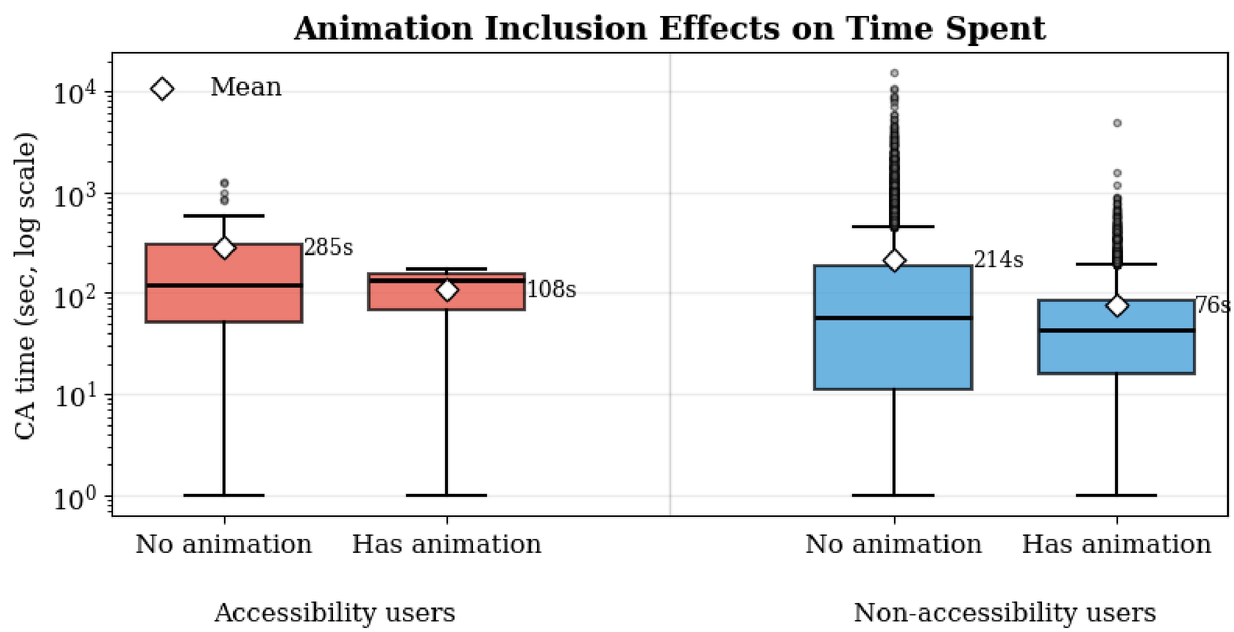

Figure 8. Time per Challenge Activity by animation inclusion and accessibility status (log scale). Diamonds indicate means.

Across major CAs, accessibility users consistently spend more time than non-accessibility users (**Figure 9**). The largest time differences occur in CodeWriting and CodeWarmup activities, while differences are smaller for more constrained interaction types such as ProblemSolving and CodeOutput. This pattern indicates that the differences in time spent are more pronounced in open-ended activity format, implying differences in performance or task difficulty.

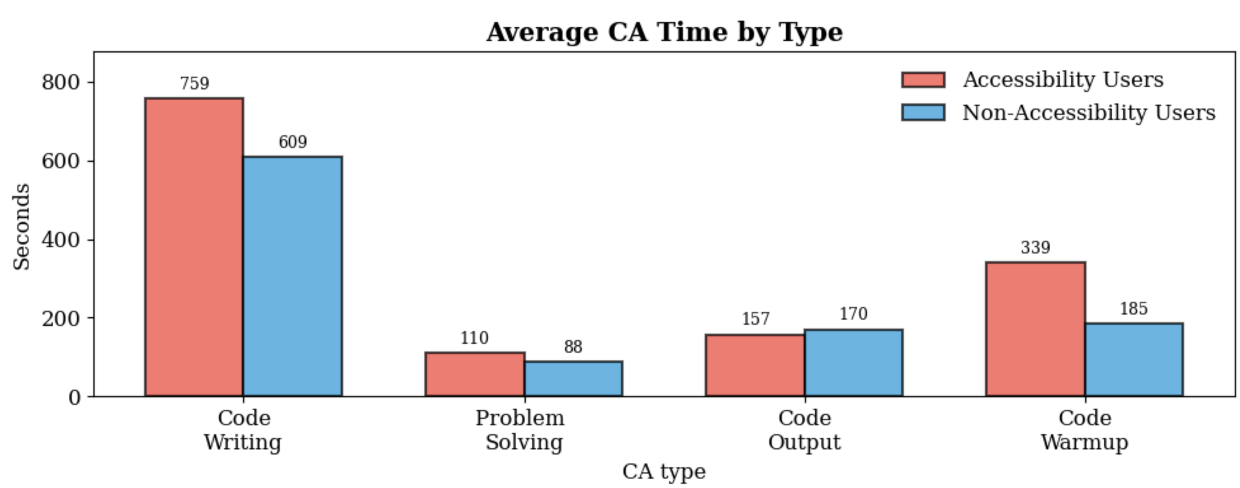

Figure 9. Mean time per Challenge Activity by activity type and accessibility status.

The following are the results of the study on platform-wide student engagement with accessibility features.

Year-to-year trends
Between 2024 and 2025, the total number of students platform-wide increased from 44,457 to 45,903. However, the percentage of students that use accessibility features decreased from 11.39% to 9.19%. (**Figure 10**)

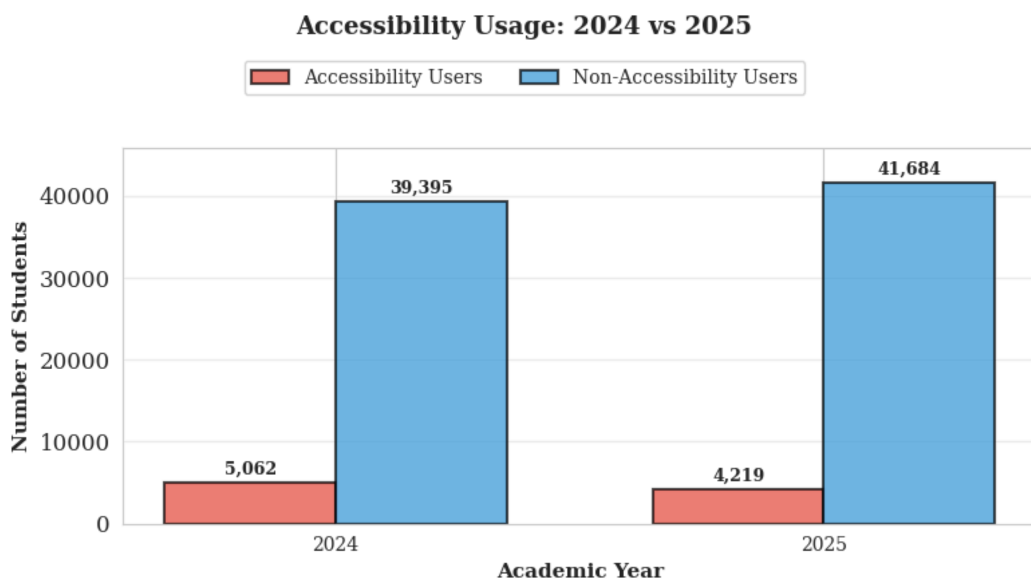

Figure 10. Number of students by accessibility status for academic years 2024 and 2025.

Class size impact
Across all classes platform-wide, very small classes have disproportionately higher accessibility ratios, indicating that students in such classes are more likely to use accessibility features. In contrast, in large classes most students do not use accessibility features, with the exception of a handful of outlier classes in which accessibility ratio reaches 50%. (**Table 2, Figure 11**)

Table 2. Percentage of accessibility users by class size category.

Class size category	Number of classes in category	Accessibility ratio (%)		
		Mean	Median	Std. Dev
Very Small (1-5)	9089	28.77	0.00	39.47
Small (6-50)	3835	7.83	0.00	13.92

Medium (51-100)	153	4.19	1.96	8.51
Large (101-150)	44	4.54	2.38	8.48
Very large (151-200)	8	9.58	4.89	15.84
Extra large (200+)	7	9.90	1.83	19.66

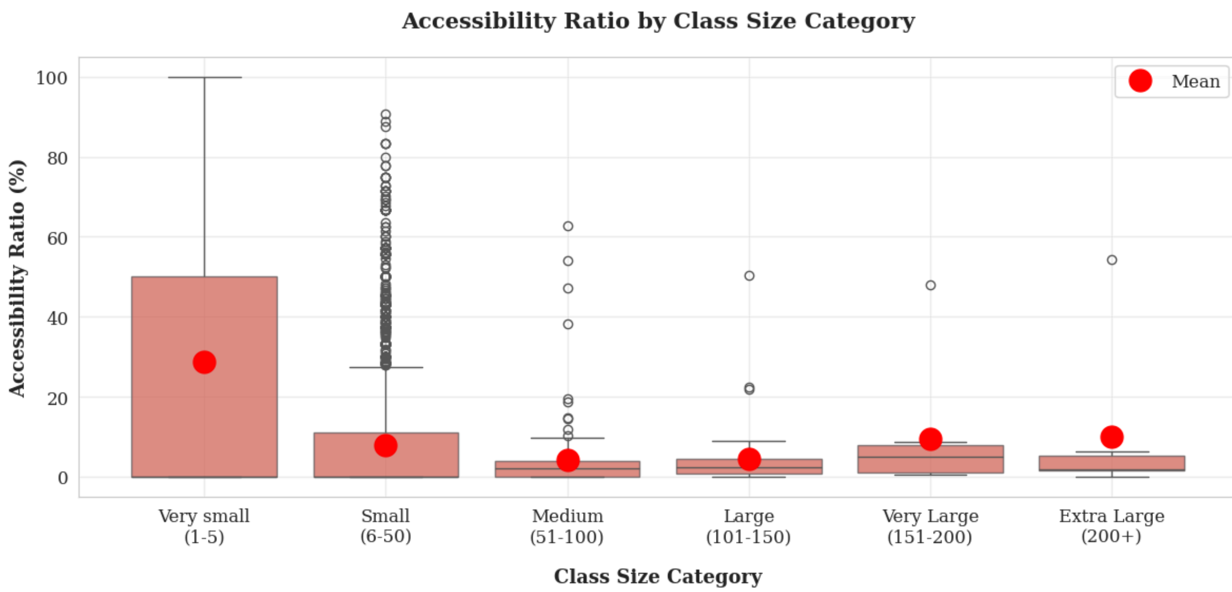

Figure 11. Distribution of accessibility ratio by class size category. Red circles indicate means.

Course level impact

Accessibility feature usage varies significantly by course level platform-wide. Introductory courses (Level 1) show the highest accessibility usage rate at 11.7% among the 37,579 students taking courses at this level. Usage drops to 9.8% at Level 2 and reaches the lowest point at Level 3 with just 6.9% of students. Level 4 and the Graduate/Other Level show moderate usage at 9.3% and 10.3% respectively. (**Figure 12**)

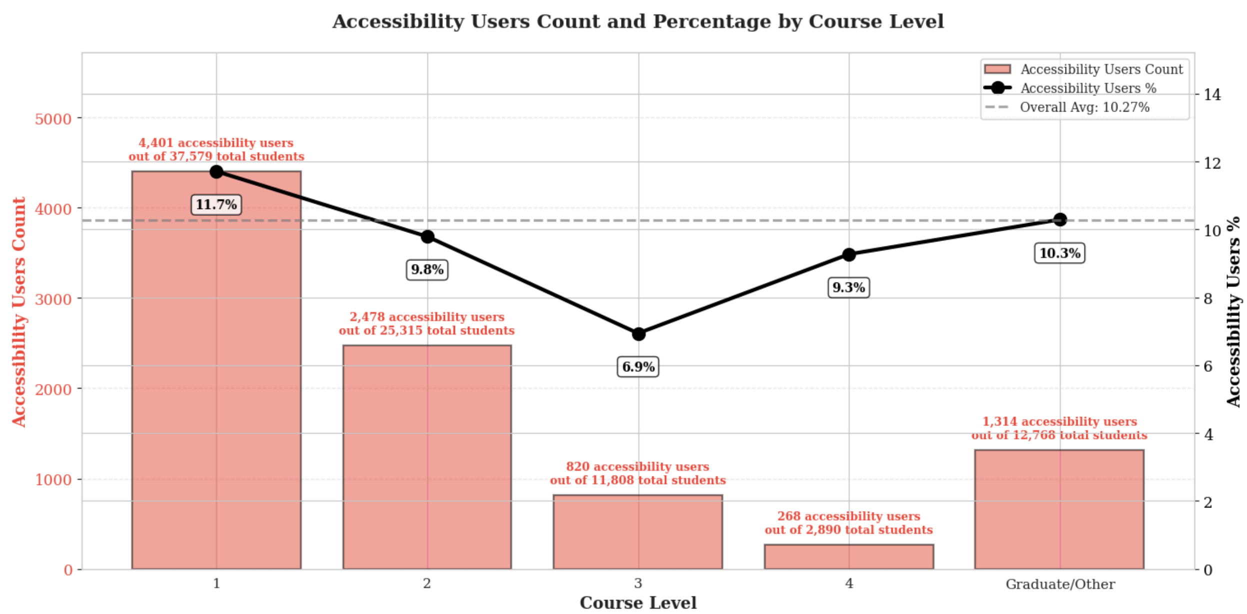

Figure 12. Accessibility users by course level: count (bars) and percentage of total enrollment (line). The dashed line indicates the overall average (10.27%).

Subject-based impact

The data shows a consistent pattern of accessibility usage across all subjects platform-wide: very small classes with 5 or fewer students have significantly higher percentage of accessibility users compared to regular classes of more than 5 students. Very small classes show usage rates ranging from roughly 18% to 36%, while regular classes typically have rates between 3% and 8%. This trend holds uniformly across all academic subjects in STEM, suggesting that class size is a strong predictor of accessibility usage regardless of subject matter. (**Table 3**)

Table 3. Distribution of students and accessibility user percentage across subject areas, comparing very small classes (≤5 students) and regular classes (>5 students).

Subject	Very small Classes (≤ 5 students)		Regular Classes (> 5 students)	
	Total students	Accessibility users %	Total students	Accessibility users %
Programming & Software Development	7910	24.7	34150	7.8
Computer Science Fundamentals	2306	25.5	12074	6.4

Algorithms, Data Structure & Discrete Math	1979	21.2	7218	4.6
Engineering Sciences	665	21.2	4793	3.9
Computer Systems & Architecture	1453	21.7	3675	4.5
Data Science & Analytics	1507	24.8	3203	6.7
Math, Statistics & Scientific Computing	967	30.8	1498	4.6
Web & Mobile Development	456	17.8	1639	4.5
Electrical & Computer Engineering	463	19.0	1591	2.6
Information Technology & Security	796	28.3	1043	5.2
Other	571	35.6	403	8.4

Student engagement through feedback

Platform-wide feedback data reveals interesting findings on the relationship between accessibility usage and feedback contribution. Among students who provided positive section ratings, accessibility users comprised 24.4% of their group, while non-accessibility users comprised only 13.8% of theirs. For negative ratings, accessibility users accounted for 11.6% compared to 7.6% for non-accessibility users. Regarding bug reports, 7.2% of accessibility users reported bugs, while only 4.3% of non-accessibility users did so. The majority of students provided no section feedback at all: 70.4% of accessibility users and 81.6% of non-accessibility users gave no feedback. These patterns show that accessibility users are consistently more engaged across all feedback types. They are more likely to provide both positive and negative ratings, more willing to report technical problems, and less likely to remain silent about their learning experience. (**Figure 13**)

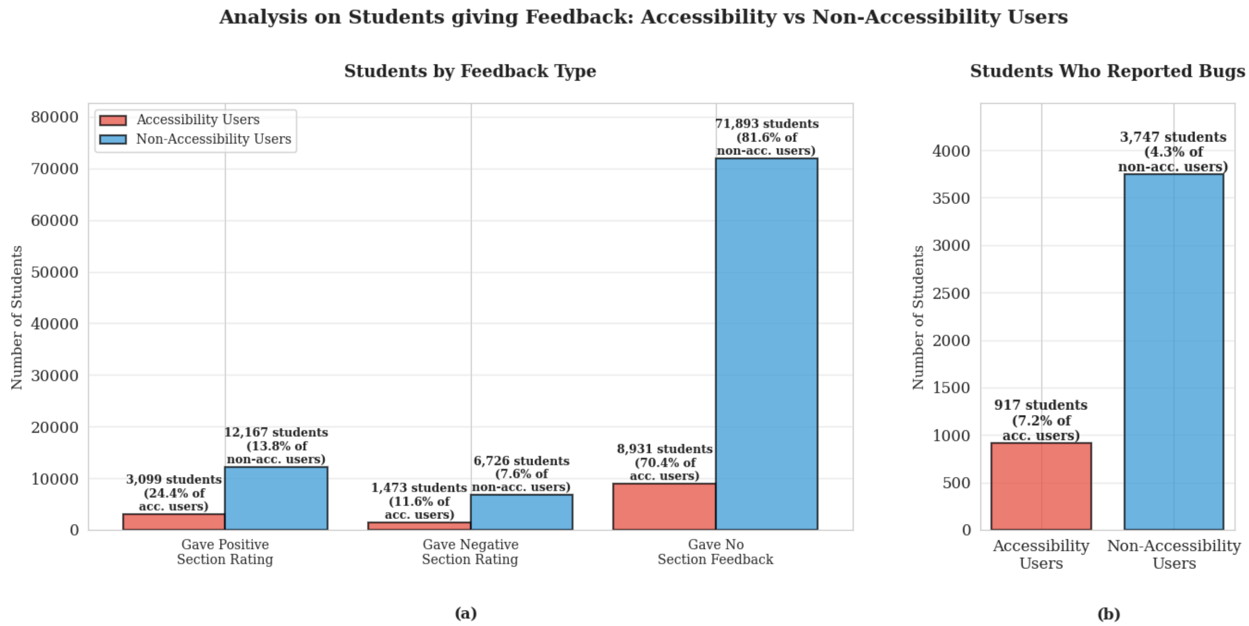

Figure 13. Comparison of feedback behavior between accessibility and non-accessibility users: (a) section rating responses by type, (b) number of students reporting bugs. Percentages indicate proportion within each user group.

Overall performance assessment

Both accessibility users and non-accessibility users demonstrate remarkably similar patterns across all performance categories platform-wide. Approximately 77-80% of students in both groups achieve excellent average CA scores of at least 95%, with fewer than 2% struggling. Accessibility users show slightly lower performance, with 77.2% in the Excellent category compared to 79.6% for non-accessibility users, and proportionally more in the other categories. This difference between the two groups is statistically significant but practically negligible given $\chi^2 = 38.12$, df = 3, $p < 0.0001$, and Cramér's V = 0.022. (**Figure 14**)

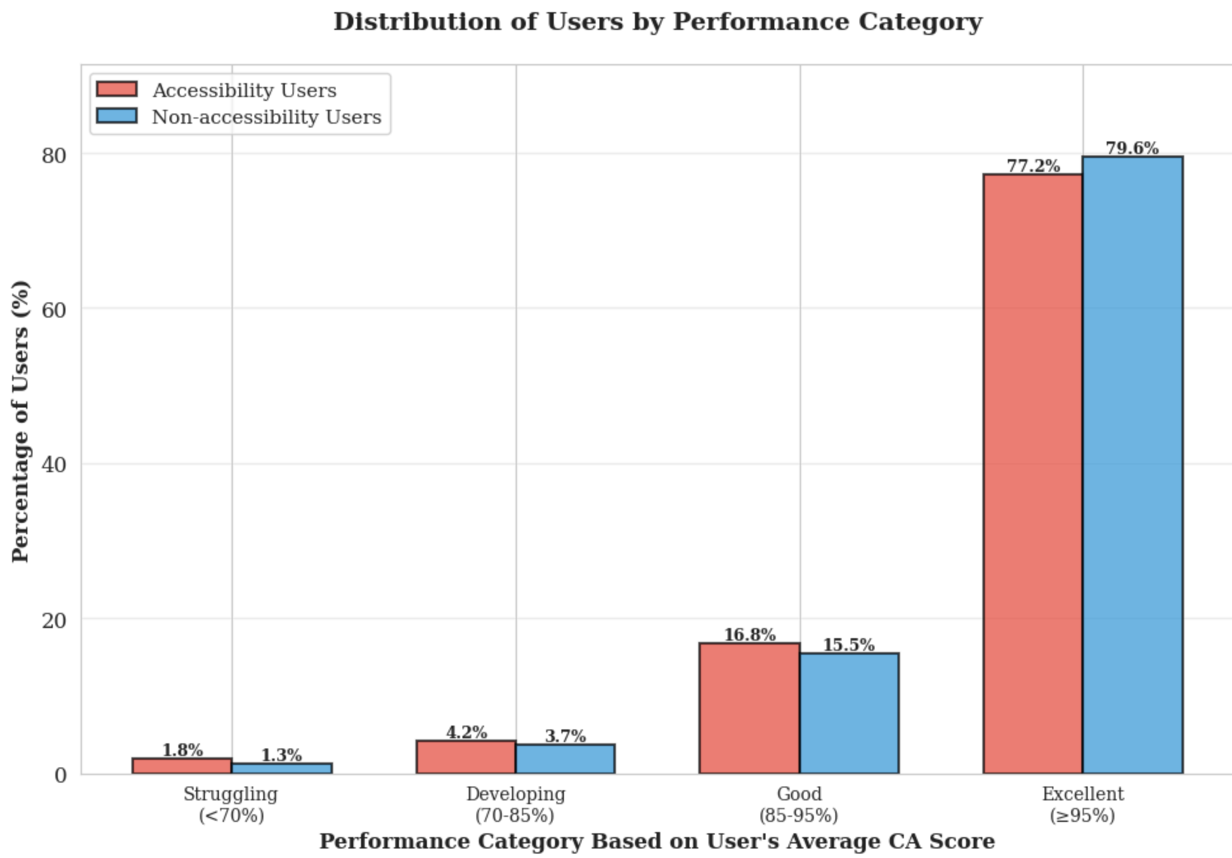

Figure 14. Percentage of accessibility and non-accessibility users in each performance category, based on average Challenge Activity scores.

Conclusion

This study shows that moderate interactivity of approximately 3-7 interactive activities per section appears to offer a practical balance, supporting engagement while avoiding the steep time increases observed in highly interactive sections. Students consistently require an average of 3 attempts to pass a CA in sections with 2-9 interactive activities, while time spent increases at approximately 100 seconds per interactive activity. Thus, sections with fewer interactive activities require proportionally less time to achieve comparable learning outcomes. However, programming topics vary in complexity. Some topics may be adequately covered with 2-3 interactive activities, while others may require 9 or more. Designers should therefore use interactive activities as the topic demands while recognizing that each additional activity in a section increases the time required for every activity in that section by approximately 100 seconds.

Approximately 10% of students enable accessibility features for their textbooks. Students in introductory classes are more inclined to use accessibility features than those taking more advanced classes in undergraduate studies. Accessibility users consistently spend more time on CAs, make slightly more CA attempts on average, and provide feedback more actively, while only marginally outperformed by non-accessibility users. These findings indicate that accessibility features support continued participation in interactive content, though the causal relationship between engagement and performance and the underlying factors driving accessibility usage require further investigation.

Future Research

While this study provides insights into interactivity and accessibility feature usage, its limitations suggest areas for future research. Data alone cannot reveal every barrier to learning. While student feedback is invaluable for identifying where to focus our improvement and accessibility efforts, only a small proportion of students provide feedback of any form. Most remain silent about their learning experience. Similarly, high completion rates do not necessarily translate to learning as struggling students could be using "gaming" behaviors, such as clicking through answers quickly to find the pattern, or using AI tools to write code for them. Furthermore, our study only captures students based on their use of accessibility features. Some students with learning disabilities, such as ADHD, dyslexia, dysgraphia, or dyscalculia, may not use accessibility features. These learners may benefit from multiple means of engagement, such as videos instead of extensive reading, interactive or guided notes, or varied instructional formats [9].

For a more comprehensive approach towards effective learning, the Center for Applied Special Technology (CAST) created the Universal Design for Learning (UDL) framework, which aims to make meaningful and challenging activities accessible for every learner [10]. Future research could apply the UDL framework to examine how the design and development of interactive instructional materials might maximize effective learning and minimize barriers for all students.

References

[1] J. Henrique Berssanette and A. Carlos de Francisco, “Active Learning in the Context of the Teaching/Learning of Computer Programming: A Systematic Review,” *Journal of Information Technology Education: Research*, vol. 20, pp. 201–220, 2021, doi: https://doi.org/10.28945/4767.

[2] S. P. Chand, “Constructivism in education: Exploring the contributions of Piaget, Vygotsky, and Bruner,” *ResearchGate*, 2024. https://www.researchgate.net/publication/378071316_Constructivism_in_Education_Exploring_the_Contributions_of_Piaget_Vygotsky_and_Bruner

[3] J. Sorva, V. Karavirta, and L. Malmi, “A Review of Generic Program Visualization Systems for Introductory Programming Education,” *ACM Transactions on Computing Education*, vol. 13, no. 4, pp. 1–64, Nov. 2013, doi: https://doi.org/10.1145/2490822.
[4] M. Messer, N. C. C. Brown, M. Kölling, and M. Shi, “Automated Grading and Feedback Tools for Programming Education: A Systematic Review,” *ACM Transactions on Computing Education*, vol. 24, no. 1, p. 3636515, Dec. 2023, doi: https://doi.org/10.1145/3636515.
[5] J. Spector, “Explorations in the Learning Sciences, Instructional Systems and Performance Technologies Series Editors,” 2011. Available: http://ndl.ethernet.edu.et/bitstream/123456789/54490/1/23.pdf
[6] R. Duran, A. Zavgorodniaia, and J. Sorva, “Cognitive Load Theory in Computing Education Research: A Review,” ACM Transactions on Computing Education, vol. 22, no. 4, Apr. 2022, doi: https://doi.org/10.1145/3483843.
[7] C. Watson and F. W. B. Li, “Failure rates in introductory programming revisited,” *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE ’14*, 2014, doi: https://doi.org/10.1145/2591708.2591749.
[8] “(PDF) Failure rates in introductory programming,” *ResearchGate*. https://www.researchgate.net/publication/220613179_Failure_rates_in_introductory_programming
[9] M. A. Pfeifer, J. J. Cordero, and J. D. Stanton, "What I wish my instructor knew: How active learning influences the classroom experiences and self-advocacy of STEM majors with ADHD and specific learning disabilities," *CBE—Life Sciences Education*, vol. 22, no. 1, 2023. https://doi.org/10.1187/cbe.21-12-0329
[10] CAST, "Universal Design for Learning Guidelines version 3.0," 2024. [Online]. Available: https://udlguidelines.cast.org/