

CQI Report: Fundamentals, Fall 2024

Submitted by:

Monika Akbar, Chair

Olac Fuentes

Daniel Mejia

Martine Ceberio

Marcelo Frias

Mahmud Shahriar Hossain

1 Subcommittee Information

The following courses are part of this subcommittee. A course may have several sections and instructors.

Subcommittee CS Courses	CRN number	Instructor name
CS 1301	12541	Daniel Mejia
CS 1101	12555, 12568, 1900, 14569, 15016	Daniel Mejia
CS 2401	11378	Monika Akbar
CS 2302	10903	Daniel Mejia

Report date: 3/18/2025 **Faculty meeting presentation date:** 5/9/2025

2 Summary

Overview of the courses and summary of reports from Fall 2024:

The fundamental courses include the three undergraduate computer science courses:

- CS 1301 (and CS 1101, lab): Introduction to Computer Science (**i.e., CS1**)
- CS2401: Elementary Data Structures and Algorithms (**i.e., CS2**)
- CS2302: Data Structures (**i.e., CS3**)

These courses previously had multiple class sections, with CS1 and CS2 also including lab sessions. Due to increased enrollment, the class sections were consolidated into a single large lecture held usually in an auditorium, supplemented by multiple lab sections for CS1 and CS2. Each lab is usually led by a graduate teaching assistant (TA) and supported by two undergraduate instructional assistants (IAs), all of whom also assist with instructional activities and management of the large classroom.

The auditorium environment has presented challenges in maintaining engagement and providing individualized student support. The instructors have been working closely with TAs and IAs to ensure students receive the necessary guidance and assistance.

In Fall 2024, all three courses had more than one hundred students. Table 1 provides a breakdown of student performance in each course, showing the number and percentage of students receiving grades. A grade of C or better is considered passing for progression to the next course in the sequence. Percentages are based on students who remained enrolled through the end of the semester and received a final grade.

Table 1: Grade distribution for CS1, CS2, and CS3

Course	A	B	C	D	F	Total
--------	---	---	---	---	---	-------

CS 1301	43 (25.1%)	27 (15.8%)	37 (21.6%)	25 (14.6%)	39 (22.8%)	171
CS 1101	51 (29.5%)	41 (23.7%)	17 (9.8%)	11 (6.4%)	53 (30.6%)	172
CS 2401	76 (48.72%)	35 (22.43%)	13 (8.33%)	4 (2.56%)	28 (17.95%)	156
CS 2302	43 (29.1%)	36 (24.3%)	35 (23.6%)	10 (6.8%)	24 (16.2%)	148

CS1 had 182 students initially enrolled, of whom 171 completed the course with a grade. Many students who failed had low attendance, often missing key exams. CS1101, which had five sections, followed a similar pattern, with students who struggled often having poor attendance. The course relied heavily on graduate TAs, assisted by undergraduate IAs, to facilitate instruction.

CS2, another large course with 156 students, faced challenges in providing individualized attention during lectures. Lab sessions were used for more personalized feedback, with students completing ten lab assignments over the semester. Assessments included midterms, quizzes, and homework, along with ungraded in-class activities designed to engage students. A free interactive eBook was used to support learning – the book offered similar, if not more, depth to zyBooks content. While 79.49% of students passed, a portion failed due to non-attendance, with 15 students missing the final exam. TA preparedness was a recurring challenge, as training them during the semester added to the complexities of managing a large class. The use of PULSE, an online coding platform, provided valuable insights into student coding practices.

CS3 enrolled 148 students, with an average attendance of 83.7%. However, some students attended infrequently and only took exams, which slightly lowered overall course performance. Despite this, the passing rate remained consistent with previous semesters and instructors. Across all three courses, attendance played a critical role in student success, and structured support from TAs and interactive learning tools helped address the challenges of large class sizes.

In terms of meeting the *learning outcomes*, in CS1, most course outcomes were successfully assessed multiple times, with comprehensive labs playing a key role. While a few topics, such as IDE usage and teamwork, were not assessed, students generally performed well, with many outcomes rated as "Good" or "Excellent." The integration of iPad notes and live coding proved beneficial.

CS2 saw strong student performance in binary trees, binary search trees, and stacks. While some struggled with strings, arrays, and recursion despite an early review, a targeted lab assignment on string operations showed positive results. Queues, introduced later in the semester, may benefit from more instructional time. Managing exam accommodations was made easier with the support of teaching assistants.

In CS3, students faced challenges with graph algorithms, which are typically covered toward the end of the course. This complex topic requires more dedicated time, and future iterations of the course will adjust to ensure better understanding. Overall, students showed progress in key areas, and refinements in pacing and instructional focus will further enhance learning outcomes.

Committee Activities and Meetings Overview

The committee met on the following dates. Each entry includes a brief summary of the discussion held on that day.

[October 22, 2024] The committee discussed whether to include Discrete Structures in data collection (we do not need to include it), differences in textbooks across CS1–CS3, and ways to gather student feedback in large classes—potentially through an existing campus survey initiative led by Dr. Ceberio. The group also debated how to calculate the percentage of students meeting learning objectives, with no final decision reached. A suggestion was made to review learning objective coverage after each exam to allow time for midsemester adjustments.

[December 4, 2024] The committee discussed a tentative timeline for completing the Fall 2024 CQI report and reviewed progress on covering all learning objectives for the current semester. Members were encouraged to organize data and sample work (high, medium, low) in designated folders. Key topics to address in the report include increased enrollment and class sizes, textbooks used in CS1–CS3, TA/IA support and training, student retention, and a proposal to add a lab component to CS3.

[March 19, 2025] The committee reviewed updates to the CQI report, recommending the addition of summary tables for CS1, CS2, and CS3 to show pass/fail/enrollment data (added Table 1). Members discussed investigating enrollment trends over time. Members suggested multiple revisions to course outcomes included modifying CS1101 outcome 2.8 to allow IDE use while emphasizing the compilation process, and removing outcome 2.9. Additional recommendations included guidance on introducing stacks and queues in CS2. The committee also proposed the addition of a 1-credit lab for CS3.

[May 9, 2025] The Fall 2024 Fundamentals CQI report was shared with the department faculty for feedback and comments.

[May 19, 2025] A follow-up meeting was held to address faculty comments from the earlier meeting on May 9th. In response to questions about the rigor of assessment instruments, the committee reviewed questions related to Level 1 and Level 2 learning outcomes in CS2 and Level 3 outcomes in CS3. The committee agreed that the questions were generally appropriate and recommended using multiple assessment instruments to evaluate each learning outcome. While labs were acknowledged as valuable for reinforcing learning, their effectiveness as stand-alone assessment tools was seen as limited due to the additional time and support available to students.

The committee also discussed the Teaching Evaluation Effort (TEval), in which three members participated by applying rubrics, reviewing course content, and conducting peer observations. The group emphasized the need for ongoing discussion around comprehensive assessment methods and proposed developing a process for continuous improvement and sharing insights with future instructors.

The committee plans to revisit the following faculty feedback in Fall 2025:

- Incorporating GenAI and ethical AI topics into Level 1 or 2 learning outcomes for CS1.
- Clarifying how to calculate the percentage of students meeting each learning outcome—whether based on all students who took the test or only those who completed the final exam.

3 Recommendations and follow up

3.1 Reflection on previous report recommendations and actions taken

For CS1301 and CS1101, no immediate actions were taken based on the previous report, as CS1301 was redesigned and evaluated by the fundamental committee, and CS1101 had no specific recommendations.

In CS2401, several recommendations were implemented. Early lab sessions were dedicated to reviewing CS1 topics, guided by a student survey to identify areas needing reinforcement. These sessions incorporated both practice activities and lab assignments, which proved effective. Regular review sessions were also introduced, with additional sessions held before midterms and finals. Regarding course materials, the Pearson eBook was not adopted due to cost concerns. Instead, the course transitioned to the free openDSA eBook, supplemented with

free Pearson resources and interactive platforms like CodingBat and PythonTutor for problem-solving and code tracing.

CS 2302 continued to emphasize in-class problem-solving and allocated additional time to complex topics such as graph algorithms. While attendance and drops were still a challenge, the course stayed focused on helping students stay engaged and succeed. Course evaluation response rates in this course were strong (70–80%), demonstrating substantial improvement in student participation for CS 2302.

3.2 Recommendations for this cycle – involving only the faculty who teach these classes.

CS1301

1. None

CS1101

1. Update Outcome 2.8 from “Instead of IDEs, use a command line interface (terminal) to compile and execute programs.” To “Demonstrate understanding of the code compilation process by compiling and executing programs using a command-line interface.”
 - a. **Rationale:** The revised wording makes this outcome less prohibitive of using IDEs while ensuring that students develop a clear understanding of the compilation process. Understanding the compilation process through the command line reinforces fundamental concepts of how code is processed, executed, and debugged beyond the abstraction of an IDE. However, students should begin to familiarize themselves with practical and real-world tool usage.
 - b. Students can still use compilation commands using most IDEs (e.g., VS Code), but this becomes an unnecessary task for many students beyond the first few weeks.
2. Remove outcome 2.9 (Use teamwork roles and strategies in the classroom)
 - a. **Rationale:** There is no practical way for this to be assessed in this course
 - b. Most (if not all) assignments are done individually, which is a key component of the course

CS2401

1. Continue having one TA and two IAs for each lab section. It would be better if we could hire PhD students as TAs. They are more experienced learners and usually have more flexibility in their schedule to accommodate some of the demands of large classes. It would be even better if we could retain some of the TAs over the semesters so that they can refine their approaches and train the new cohort of TAs and IAs.
2. Continue with the openDSA eBook. It seems like a good alternative to some of the books we tried earlier.
3. In the future, we may explore introducing queues and stacks conceptually (e.g., breadth-first traversal using a queue) earlier in the semester, after covering linked lists and trees, with a focus on their behavior rather than implementation details. A deeper look into queues and stacks later in the semester may be more effective for building proficiency in these topics. Since pre-order and postorder traversals naturally align with stack-based approaches, incorporating stack usage when covering these traversals could also enhance understanding.

CS2302

1. Outcomes that were not met include Level 2.3 (a) through (f): *Describe, implement, and apply the following graph algorithms: Breadth-first search, Depth-first search, Topological sorting, Minimum spanning trees (Kruskal's and Prim's), Single-source shortest paths (Dijkstra's algorithm).*
 - a. This is typically the last section of the course, as such it may not get the ample time it needs. It is also a notoriously difficult topic for many students. More effort will be done to increase the amount of time spent on this section.
2. Additional time will need to be spent on various outcomes described in level 1.
3. Adding a 1-credit lab to the course, which would include 3 contact hours per week. Currently, CS3 does not have a lab component, and this addition could provide students with hands-on practice to reinforce key concepts covered in lectures.

3.3 Recommendations for this cycle that require departmental approval (e.g., changes in outcomes)

CS1101

1. Update Outcome 2.8 from “Instead of IDEs, use a command line interface (terminal) to compile and execute programs.” To “Demonstrate understanding of the code compilation process by compiling and executing programs using a command-line interface.”
 - a. **Rationale:** The revised wording makes this outcome less prohibitive of using IDEs while ensuring that students develop a clear understanding of the compilation process. Understanding the compilation process through the command line reinforces fundamental concepts of how code is processed, executed, and debugged beyond the abstraction of an IDE. However, students should begin to familiarize themselves with practical and real-world tool usage.
 - b. Students can still use compilation commands using most IDEs (e.g., VS Code), but this becomes an unnecessary task for many students beyond the first few weeks.
2. Remove outcome 2.9 (Use teamwork roles and strategies in the classroom)
 - a. **Rationale:** There is no practical way for this to be assessed in this course
 - b. Most (if not all) assignments are done individually, which is a key component of the course

CS 2302

1. Explore the possibility of adding a 1-credit lab to the course, which would include 3 contact hours per week. Currently, CS3 does not have a lab component, and this addition could provide students with hands-on practice to reinforce key concepts covered in lectures.

4 Individual CS Course Outcomes Reports

[CS1301 CQI Course Report FA24 Mejia](#)

[CS1101 CQI Course Report FA24 Mejia](#)

[CS2401 CQI Course Report FA24 Akbar](#)

[CS 2302 CQI Course Report FA24 Mejia](#)