











that the student demand was met. The exact class composition for each quarter is shown in Figure 1.

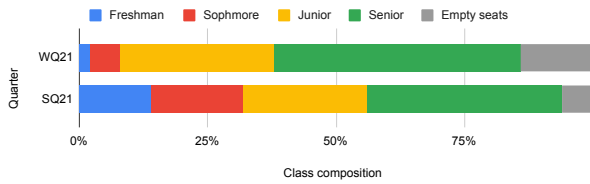


Figure 1: Class composition per course offering

The course was intrinsically designed to be accessible as soon as after CS1. We were surprised when the first offering counted an overwhelming 90% of upper-division students, including 56% of seniors alone. That said, it confirmed two of our initial assumptions which motivated the creation of this course: (1) students recognize the importance of computer ecosystem literacy for their professional success, especially as they are about to graduate, and (2) students feel like this literacy is not being sufficiently covered during their undergraduate education. Such a big share of upper-division students is also likely due to the fact that they have priority for registration.

To ensure that the second offering would be more balanced, we delayed enrollment until the open registration period, during which students of all levels have equal access to available courses. While seniors still held the most significant enrollment, the share of lower-division students dramatically increased (from 9% to 37%).

We are confident that as our course keeps being regularly offered, the proportion of lower-division students will steadily increase and soon become the majority, as we had initially expected.

### 5.2 Student feedback

At the end of each quarter, we asked students to complete an optional survey, to gather feedback about their experience. Unfortunately, the survey completion rate for the first class was relatively low, with only about 28% of students responding. For the second class, we gave a tiny amount of extra participation credits for completing the survey which boosted the completion rate to about 60%. All the results shown in this subsection represent the cumulative percentage across the two classes.

The main question that our survey aimed to investigate was whether students had increased their knowledge and self-efficacy in the topics covered by the course. We first asked them to rate how familiar with these topics they felt prior to taking our course on a 4-point Likert scale. As shown in Figure 2, the results unsurprisingly matched our original February 2020 survey, which prompted the creation of this course; for most topics, a majority of students reported having a low level of familiarity or no familiarity at all.

As shown in Figure 3, we then asked them to rate their perceived progress after having completed the course, also on a 4-point Likert scale. The vast majority of students reported a high or moderate level of progress across all the course topics.

In addition to evaluating students' overall progress in the course topics, we finally wanted to determine to what extent they left the course with a growth mindset. We therefore asked students if

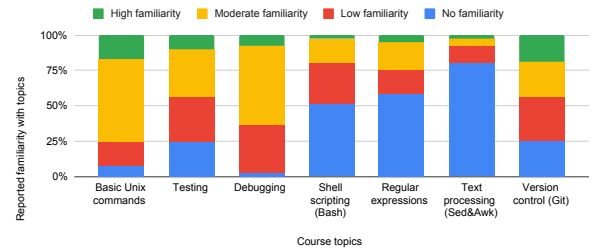


Figure 2: Student familiarity with course topics prior to taking class

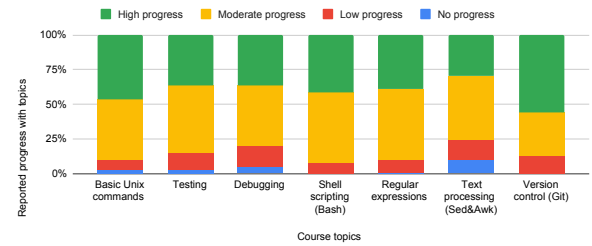


Figure 3: Student progress on course topics after taking class

they had incorporated some of these new software development practices into their workflow and if they had become more curious about CLI environments and tools. As shown in Figure 4, the vast majority of students reported that they had, for both metrics.

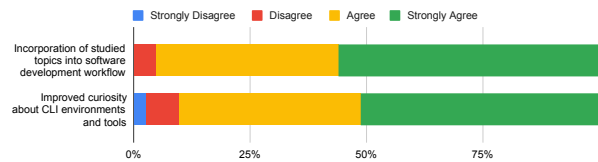


Figure 4: Growth mindset after taking class

## 6 CONCLUSION AND FUTURE WORK

In this paper, we described "The Missing CS Class," a student-led undergraduate course that covers topics aiming to increase students' computing ecosystem literacy and therefore helps reduce the academia-industry gap. We provided a detailed description of the course contents, which we hope others can adapt to their needs. The very positive feedback we received from students indicates that we accomplished our objectives of inspiring curiosity and encouraging independent learning.

Our long-term objective is now for this course to continue as a generational, student-led effort that can stand on its own after we graduate. We are currently looking into a rolling approach in which formerly enrolled students later become the ones teaching the course. Finally, to benefit the community at large, we have published all the lecture materials online at <https://missing.cs.ucdavis.edu>.

## REFERENCES

- [1] Anish Athalye, Jon Gjengset, and Jose Javier Gonzalez. January 2019-20. The Missing Semester of Your CS Education. Massachusetts Institute of Technology. Retrieved 2021/08/11 from <https://missing.csail.mit.edu/>
- [2] Fabrice Bellard et al. 2000. Ffmpeg. Retrieved 2021/08/11 from <https://www.ffmpeg.org/>
- [3] Adam Blank, Josh Zimmerman, and Jake Zimmerman. Fall 2020. 07-131 – Great Practical Ideas in CS. Carnegie Mellon University. Retrieved 2021/08/11 from <https://www.cs.cmu.edu/~07131/f20/>
- [4] Michael Chang, Julie Zelenski, Chris Gregg, and Nick Troccoli. Spring 2021. Assignment 0: Intro to Unix and C. In *CS107: Computer Organization & Systems*. Stanford University. <https://web.stanford.edu/class/archive/cs/cs107/cs107.1216/assign0/> (2021/08/11).
- [5] CC2020 Task Force. 2020. *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA.
- [6] Paul Hilfinger. Spring 2021. Lab 6: Unit Testing and Integration Testing for Enigma. In *CS 61B: Data Structures*. University of California, Berkeley. Retrieved 2021/08/11 from <https://inst.eecs.berkeley.edu/~cs61b/sp20/materials/lab/lab6/index.html>
- [7] Vidar Holen et al. 2013. ShellCheck - shell script analysis tool. Retrieved 2021/08/11 from <https://www.shellcheck.net>
- [8] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery, New York, NY, USA.
- [9] Tyson Kendon and Ben Stephenson. 2016. Unix Literacy for First-Year Computer Science Students. In *Proceedings of the 21st Western Canadian Conference on Computing Education* (Kamloops, BC, Canada) (WCCCE '16). Association for Computing Machinery, New York, NY, USA, Article 14, 4 pages. <https://doi.org/10.1145/2910925.2910930>
- [10] Chen Li, Emily Chan, Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2019. *Towards a Framework for Teaching Debugging*. Association for Computing Machinery, New York, NY, USA, 79–86. <https://doi.org/10.1145/3286960.3286970>
- [11] Steven Lu and Kiri L. Wagstaff. 2020. *MSL Curiosity Rover Images with Science and Engineering Classes*. <https://doi.org/10.5281/zenodo.3892024>
- [12] Renée Mecauley, Sue Fitzgerald, Gary Lewandowski, Laurie Murphy, Beth Simon, Lynda Thomas, and Carol Zander. 2008. Debugging: A review of the literature from an educational perspective. *Computer Science Education* 18 (06 2008). <https://doi.org/10.1080/08993400802114581>
- [13] Nick Parlante. Fall 2007-08. Unit Testing. In *CS108: Object Oriented Programming*. Stanford University. <https://web.stanford.edu/class/archive/cs/cs108/cs108.1082/handouts/06UnitTesting.pdf> (2021/08/11).
- [14] Alex Radermacher and Gursimran Walia. 2013. Gaps between Industry Expectations and the Abilities of Graduates. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) (SIGCSE '13). Association for Computing Machinery, New York, NY, USA, 525–530. <https://doi.org/10.1145/2445196.2445351>
- [15] Alex Radermacher, Gursimran Walia, and Dean Knudson. 2014. Investigating the Skill Gap between Graduating Students and Industry Expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) (ICSE Companion 2014). Association for Computing Machinery, New York, NY, USA, 291–300. <https://doi.org/10.1145/2591062.2591159>
- [16] Alex D. Radermacher. 2012. *Evaluating the gap between the skills and abilities of senior undergraduate computer science students and the expectations of industry*. Ph.D. Dissertation. <https://www.proquest.com/dissertations-theses/evaluating-gap-between-skills-abilities-senior/docview/1018399283/se-2?accountid=14505> Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-25.
- [17] Vennila Ramalingam, Deborah LaBelle, and Susan Wiedenbeck. 2004. Self-Efficacy and Mental Models in Learning to Program. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (Leeds, United Kingdom) (ITiCSE '04). Association for Computing Machinery, New York, NY, USA, 171–175. <https://doi.org/10.1145/1007996.1008042>
- [18] Nicholas Weaver. Spring 2019. Lab 0: Setup. In *CS 61C: Great Ideas in Computer Architecture (Machine Structures)*. Stanford University. <https://inst.eecs.berkeley.edu/~cs61c/sp19/labs/lab0.pdf> (2021/08/11).