A Case Study of Taking AP Computer Science Principles: A Student's Perspective

Sarah Cameron
Department of Computer Science
University of West Florida
Pensacola, FL, USA
sec80@students.uwf.edu

Tony Pham
Department of Computer Science
University of West Florida
Pensacola, FL, USA
hdp12@students.uwf.edu

Dr. Sikha Bagui
Department of Computer Science
University of West Florida
Pensacola, FL, USA
bagui@uwf.edu

ABSTRACT

With the increased demand for Computer Science degrees in the work force, Computer Science is becoming more prominent in high schools. AP Computer Science Principles (AP CSP) is a course that serves as a bridge into Computer Science. Code.org provides a year-long curriculum for this AP course to be led by teachers in the classroom. Beyond an analysis of the pass rates of students, and with the recency of the AP CSP course, a reflection of the AP CSP curriculum from the student's perspective is in order. This study breaks down the strengths and weaknesses of AP CSP from a student's perspective. Results show there are many strengths compared to weaknesses in relation to the Code.org curriculum. However, the course can be a little challenging in motivating and engaging students if not executed properly by the teacher.

ACM Reference format:

Sarah Cameron, Tony Pham and Sikha Bagui. 2023. A Case Study of Taking AP Computer Science Principles: A Student's Perspective. In *Proceedings of 2024 Special Interest Group Computer Science Education (SIGCSE'24), March 20-23, Portland, OR.* ACM, Portland, OR, USA, 3 pages. https://doi.org/10.1145/3626253.3635490

1 INTRODUCTION

The field of computer science (CS) is a growing field, in need of graduating students with CS degrees to fill positions. In Florida alone, there were 28,088 open positions with only 3,808 graduates in 2022 [1]. To encourage students to pursue CS degrees at the college level, CS education is being introduced to students at the high school level. Many high schools are introducing CS courses, one version being the Advanced Placement (AP) Computer Science Principles (CSP) course. AP CSP is a course offered to high schoolers that allows them to earn college credit and learn about CS in a broader scope. Due to the recency of the course, research relating to the course is just beginning to be published. One area of lacking research is insight into the student's perspective on the course and its content. This work intends to present a reflection of the AP CSP curriculum from the student's perspective.

SIGCSE 2024, March 20–23, 2024, Portland, OR, USA

ACM ISBN 979-8-4007-0424-6/24/03. https://doi.org/10.1145/3626253.3635490

2 BACKGROUND INFORMATION

The AP CSP course was first launched in the 2016 exam season with the intent of bridging the gap for underrepresented groups in computer science including women, African Americans, and Hispanics. This intent has shown mixed results. AP's own report showed positive increases in underrepresented student enrollment in AP STEM-related courses [2], but this contrasts with data showing that the AP CSP course compared to the AP Computer Science A (CSA) course correlated with fewer declared CS majors (16.9% versus 28.3% respectively) [3]. A state-wide study on the AP CSP course, conducted in the state of Maryland, saw a decrease in the offerings of AP CSA once AP CSP was introduced in many districts [4]. Overall, research shows a mixed conversation about the AP CSP course, with little to no research specifically focusing on the student's perspective on the course.

3 UNDERSTANDING CODE.ORG

The students related to this research were taught using the AP CSP Code.org curriculum, a website-based curriculum recognized by AP [5]. The curriculum incorporates the 5 major topics of the exam into a 10-unit course with a combination of coding and conceptual-centric units.

The Code.org curriculum employs various resources to help students and teachers understand the course including worksheets, multi-media presentations, videos, projects, and assessments. The coding-centric units also employ a model by Code.org called EIPM (Explore, Investigate, Practice, Make). This EIPM model is intended for teachers to build a strong foundation of knowledge with their students before slowly allowing students to take charge of each topic [6].

4 ANALYSIS

To assess the Code.org curriculum for AP CSP from a student's perspective, both strengths and shortcomings of the course will be highlighted. Alongside these strengths and weaknesses, a section discussing the students' direct experiences is included.

4.1 Strengths

4.1.1 Usage of Multimedia Presentations. Code.org uses multimedia presentations to engage, motivate, and develop students' understanding of the material. This style of presentation is beneficial for students as it can "stimulate cognitive aspects of learning… [and] increase student motivation. [7]" Code.org directly integrates AP CSP vocabulary into multimedia presentations. Studies have shown that multimedia integration of vocabulary have helped in learning and retaining more vocabulary [8, 9].

4.1.2 Skill-Building: Collaboration to Debugging, Skills of Computer Science. Debugging, pair programming, and general collaboration are all skills emphasized in the course through projects and lessons. Debugging and collaborative skills are both vital in the field of computer science. A study has shown that learning through debugging allows students to have a more thorough understanding of coding principles [10]. The pair programming model has been shown to improve the confidence of students and the quality of projects when used in classes [11].

4.1.3 Projects, Assessments, and Handouts: Skill Assessment Towards Success. Code.org has a heavy emphasis on hands-on projects, AP-relevant assessments, and assistive handouts. A longitudinal study showed that students can develop a conceptual understanding rooted in creative and deeper thinking through projects [12].

4.2 Shortcomings

4.2.1 Conceptual and Coding Units: An Issue of Pacing. One major issue seen throughout the course is the issue of pacing. Units focusing on concepts relating to CS including cybersecurity, networks, and global connection are slow in nature compared to the extremely fast-paced coding units. Units relating to coding are often taught so quickly that newer students are unable to keep up.

4.2.2 Projects and Assessments: Disconnected from Curriculum. The projects and assessments for some units can feel disconnected from the material taught through the presentations. This disconnect doesn't allow students to benefit from the creative learning style brought about by project-centric learning.

4.3 Student's Perspective

4.3.1 Resources and Presentations: The Good and The Bad. The resources provided by Code.org for teachers to use are extensive and intended to support even the newest computer science educators in helping their students. However, these resources can fail to be effective for students if teachers do not provide access to these resources.

4.3.2 The EIPM Model: A System Failed in Application. The EIPM Model in design is meant to encourage students to slowly take initiative in their learning pertaining to each coding topic. However, if teachers fail to embrace the model's intended principles of weaning students off teacher support with each topic, the model falls flat. Failure to embrace the model by students and teachers can also harm the learning environment for coding-related concepts.

4.3.3 Teacher Involvement: A Curriculum Made for the Engaged. The curriculum is thorough in its explanations, providing extensive notes and a tight-knit community for teachers to connect to, but oftentimes, teachers don't embrace these available sources.

5 CONCLUSIONS

In general, beneficial approaches to teaching are found through the AP CSP Code.org curriculum including multimedia presentations with vocabulary integration; skill building including debugging, pair programming, and collaboration; and emphasis on teacher involvement including guidance and feedback. Despite the strong course, there are still some weak areas of pacing and disconnected projects and assessments. A big part of the student feedback emphasized the strength of the course material and setup but the failure of the teachers to embrace the material.

Overall, the students agreed that the course was quite effective at the material it taught but only failed in application due to teachers not being able to embrace all the material available.

ACKNOWLEDGMENTS

This work has been supported by NSF grant no. 2122393 and by The University of West Florida's Office of Undergraduate Research.

REFERENCES

[1] Code.org. 2022. Florida 2022 State of CS report | CS advocacy. Retrieved 2022 from https://advocacy.code.org/stateofcs.

[2] Jeff Wyatt, Jing Feng, and Maureen Ewing. 2020. AP computer science principles and the computer science pipelines. (December 2023). Retrieved 2022 from https://apcentral.collegeboard.org/media/pdf/ap-csp-and-stem-cs-pipelines.pdf.

[3] Linda J Sax, Kaitlin N. S Newhouse, Joanne Goode, Max Skorodinsky, Tomoko M Nakajima, and Michelle Sendowski. Does AP CS Principles Broaden Participation in Computing? February 2020. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education.* 542-548. https://doi.org/10.1145/3328778.3366826

[4] Heather Killen, David Weintrop, and Megean Garvin. AP Computer Science Principles' Impact on the Landscape of High School Computer Science using Maryland as a Model. February 2019. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education.* 1060-1066. https://doi.org/10.1145/3287324.3287356

[5] Code.org. 2023. CS Principles | Code.org. Retrieved 2022 from https://code.org/educate/csp.

[6] Code.org. 2022. EIPM: A Short Introduction - Google Docs. Retrieved 2022 from https://docs.google.com/document/d/1ncil5b0yWAN4LCyOeXwYuNrNKEHtN4nmAd2o-_K5Psw/preview.

[7] Iryna, Kotiash, Iryna Shevchuk, Maksym Porysonok, Iryna Matviienko, Mykyta Popov, Vitalii Terekhov, and Oleksandr Kuchai. Possibilities of using multimedia technologies in education. June 2022. *International Journal of Computer Science and Network Security* 22, 6 (June 2022), 727-732. https://doi.org/10.22937/IJCSNS.2022.22.6.91

[8] Hamidreza Khiyabani, Behzad Ghonsooly, and Zargham Ghabanchi. Using multimedia in teaching vocabulary in high school classes. January 2014. *Journal of Advances in English Language Teaching* 2, 1 (Jan. 2014), 1-13.

[9] Bagui, S. Reasons for Increase in Learning with Multimedia. The Journal of Multimedia and Hypermedia, 7(1), (1999), 3-18.

[10] J.M. Griffin. Learning by taking apart: deconstructing code by reading, tracing, and debugging. September 2016. In *Proceedings of the 17th Annual Conference on Information Technology Education.* 148-153. https://doi.org/10.1145/2978192.2978231

[11] C. McDowell, L Werner, H. E. Bullock, J. Fernald. The impact of pair programming on student performance, perception and persistence. May 2003. In *Proceedings of the 25th International Conference on Software Engineering.* 602-607. httsp://doi.org/10.1109/ICSE.2003.1201243

[12] Dimitra Kokotsaki, Victoria Menzies, and Andy Wiggins. Project-based learning: A review of the literature. July 2016. *Improving Schools* 19, 3 (July 2016), 267-277. httsp://doi.org/10.1177/1365480216659733