Approaches to Broadening Participation with AP Computer Science Principles

Audrey Le Meur

Outline

- What is AP Computer Science Principles (AP CSP)?
- How can we evaluate AP CSP curriculum?
  - AP Scores
  - Self-efficacy/Confidence
  - Belongingness/Identity
  - Persistence/Interest
- Social Approaches
  - Supporting Students through Peer Learning Communities
  - Encouraging Cooperative Learning
- Curricular Approaches
  - Finding the Beauty and Joy in Computing
  - Engaging Students through Mobile Computing
- What can we learn about how to teach AP CSP?
- What next?

What is AP Computer Science Principles?

- Part of the series of “Advanced Placement” curriculums and exams offered by the College Board
- Developed with the specific intent of being more accessible to diverse groups than AP Computer Science A (Kick and Trees 2015)
- Introductory course with no prior knowledge required
    - Students first experience with CS

What content is covered by AP CSP?

- Computational Solution Design
- Algorithms and Programming Development
- Abstraction
- Code Analysis
- Computing Innovations
- Responsible Computing

How do teachers use AP CSP?

- The AP CSP framework defines what students should learn but not how
- Teachers have a lot of freedom including what programming language to use
- In this presentation, we look at how these choices improve student learning and have the potential to broaden participation in computing

How can we evaluate AP CSP approaches?

AP Scores

- Nationally standardized exam given in May
- Students’ exam scores are determined by:
  - 70% multiple choice exam
  - 30% “Create performance task”
    - Group programming project
    - Individual written questions
    - Individual video explanation
- Students are given a final score on a scale from 1-5

Self-efficacy and confidence

- Self-efficacy: “people’s beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives (Bandura 1994)”
  - In this case, computer science or computational thinking
- Confidence: “strength of self-efficacy (Compeau and Higgins 1995)”
  - How strongly people feel that they can complete a task

Belongingness and Identity

- Belongingness: the “feeling that you fit in and there are others like you in CS (Mark and Klein 2019)”
- We can ask several questions to get a sense of students perceptions of their own identity in CS:
  - What does the ideal CS professional look like (in comparison to you)? (McDonald et al. 2019)
  - Can someone of your race and/or gender be successful in CS? (Escobar et al. 2021)

Persistence and Interest

- Persistence: whether students want to continue studying or doing CS.
- Interest: whether students are interested in CS content

AP CSP Approaches

- Social Approach: an approach in which the researchers hope to improve outcomes by changing the way that students interact with others within the classroom
- Curricular Approach: an approach in which the content of the course is designed to be appealing and inclusive of a diverse audience

Social Approaches

Supporting Students through Peer Learning Communities (Escobar et al.)

- Recruited 40 young Black women enrolled in an AP CSP course
- 5 days plus 2 more days of summer enrichment
  - Previewed CSP concepts
  - Social activities
  - Opportunities to meet Black female role models
- Moodle site and events to meet with other students throughout the year
- Evaluations (given by pre/post survey):
  - Self-efficacy (Weese and Feldhaussen)
  - Gender and Racial Attitudes Toward Computing inventory
  - CS Professional Identity Overlap (McDonald et al.)

Results (Escobar et al.)

- 87.5% passed the AP exam
  - Higher than national pass rate for all students, male students, White male students, and Black male students
- Exam scores were positively correlated with attended PLC sessions
- Increases in self-efficacy for some skills:
  - Algorithmic thinking
  - Control flow
- But not for others:
  - Importance of computing
  - Organizing complex tasks

Results (Escobar et al.) cont.

- More positive attitudes about the ability of people from all racial backgrounds and women to succeed in CS
- Increased self-identification with their personal image of an ideal CS professional
- 59% intended to major/minor in CS in college

Encouraging Cooperative Learning (Gray et al.)

- Cooperative Learning (CL) means (Kagan and Kagan 2009):
  - Positive interdependence
  - Individual accountability
  - Equal participation
  - Simultaneous interaction
- Examples of CL structures:
  - Pair Programming (Gray et al. 2019)
  - Round Robin (Kagan and Kagan 2009)
  - Debate Team Carousel (Kagan and Kagan 2009)
  - Jot Thoughts (Kagan et al. 2015)

Methods (Gray et al.)

- Professional development for 27 teachers
- Teacher CL-use survey
- Student computing self-efficacy pre/post survey (Compeau and Higgins)

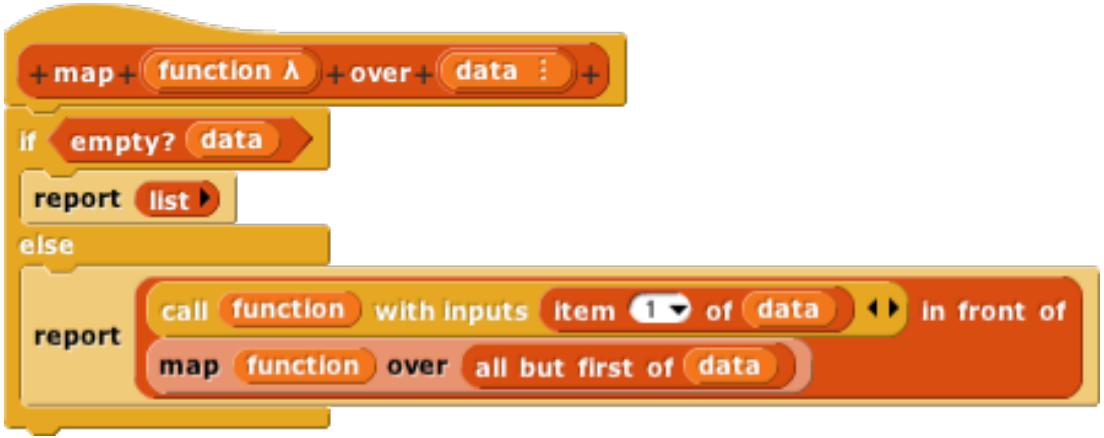
Results (Gray et al.)

- 76.6% pass rate (vs. 72.3% national pass rate)
- Use of CL only predicted higher AP scores in classes where the teacher had been teaching CS for less than three years
- No significant gains in self-efficacy over the course

Curricular Approaches

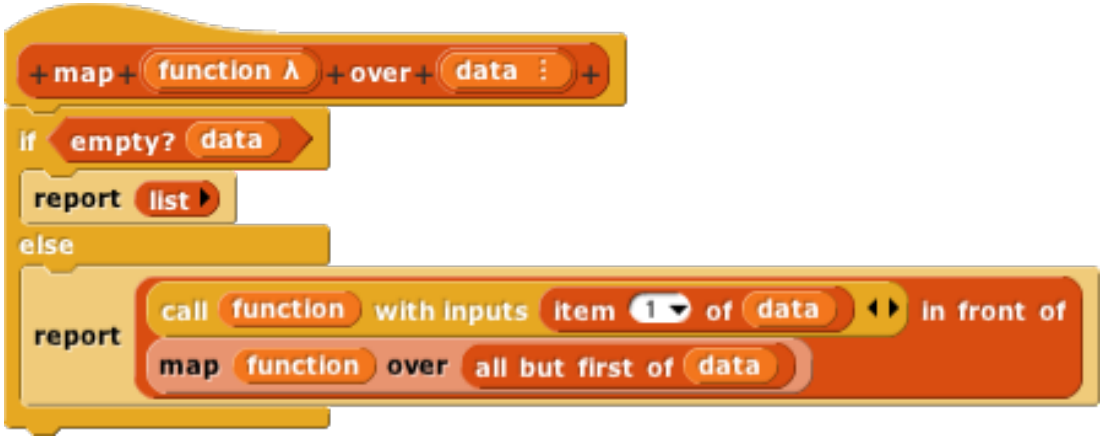
Finding the Beauty and Joy in Computing (BJC) (Goldenberg et al.)

- Aims to foster competence, confidence, and creativity
- Uses a visual programming language



Finding the Beauty and Joy in Computing (BJC) (Goldenberg et al.) cont.

- Snap*!* Allows for recursion, higher-order functions, complex data structures, object oriented programming, and lambda expressions



Finding the Beauty and Joy in Computing (BJC) (Goldenberg et al.) cont.

- Students learned about and considered the social implications of technology

Methods (Mark et al.)

- Professional development for teachers who taught 311 students in 24 NYC high schools
- CS attitude pre/post survey (Hoegh and Moskal, Lewis et al, Williams et al.)
  - Confidence, interest, belongingness, identity
- Received AP score data from NYC DOE on all students who took the AP CSP exam

Results (Mark et al.)

- 67.2% of BJC students passed (vs. 72.8% of non-BJC students)
- Removing two schools that were outliers:
  - 54.2% of BJC students passed (vs 37.7% of non-BJC students)
- Students saw significant gains in:
  - Confidence
  - Identity
- But not in:
  - Belongingness
  - Interest
- No difference in survey results between female and underrepresented minority students and male and non-underrepresented minority students

Engaging Students through Mobile Computing (Hoffman et al.)

- Based on the Mobile CSP curriculum
- Learning AP CSP concepts through mobile app design and programming
  - Use of App Inventor (Wolber), a visual programming language
- Completed the “Create performance task” by creating a mobile app
  - Encouraged to build an app that would be “socially useful”

Methods (Hoffman et al.)

- 275 teachers received about 100 hours of professional development
- Student completed pre/post survey about their attitudes and interest in CS

Results (Hoffman et al.)

- 78% pass rate on 2017 AP exam (vs. 74% nationally)
- 76% pass rate on 2018 AP exam (vs. 69% nationally)
- Women, Hispanic/Latino and multiracial students performed better than national average both years
- Black/African American students performed better only on 2017 exam.

Results (Hoffman et al.) cont.

- Proportion of groups who expressed more interest as result of course:
  - 59% of all students
  - 56% of female students
  - 56% of Black students
  - 66% of Latino students
- Proportion of groups who expressed a desire to continue doing CS (majoring in CS or pursuing CS as a career)
  - 64% of all students
  - 62% of underrepresented minority students
  - 48% of female students

What can we learn about how to teach AP CSP?

Approach	Exam Results	Self-efficacy and Confidence	Belongingness and Identity	Persistence and Interest
Social Approaches				
Supporting Students through Peer Learning Communities [7]	✓	✓*	✓	✓
Encouraging Cooperative Learning [9]	✓*	✗	N/A	N/A
Curricular Approaches				
Finding the Beauty and Joy in Computing [8, 17]	✓	✓	✓*	✗
Engaging Students through Mobile Computing [5]	✓	N/A	N/A	✓*

What next?

- Individual teachers can use one or more of these approaches in their classroom to better support students
- Districts can use these approaches to better support their teachers in teaching diverse populations

- Researchers can begin to look at the long-term effects of CSP on the CS pipeline including the effects of different CSP curriculum
- There also needs to be more work on commercially developed curriculums
    - E.g. Apple Develop in Swift, Microsoft MakeCode, Carnegie Learning Zulama

References

Unit 8 lab 4: Building higher order functions. https://bjc.edc.org/bjc-r/cur/programming/8-recursive-reporters/4-building-higher-order-functions/2-generalizing-the-map-pattern.html?topic=nyc_bjc%2F8-recursive-reporters.topic&course=bjc4nyc.html&novideo&noassignment

2020. AP Computer Science Principles: Course and Exam Description. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf

2022. About AP scores. https://apstudents.collegeboard.org/about-ap-scores

Albert Bandura. 1994. Self-efficacy. *Encyclopedia of human behavior 4* (1994), 71–81.

Deborah R Compeau and Christopher A Higgins. 1995. Computer Self-Efficacy: Development of a Measure and Initial Test, 189-211 pages. Issue 2.

References cont.

Martha Escobar, Jeff Gray, Kathleen Haynie, Mohammed A. Qazi, Yasmeen Rawajfih, Pamela McClendon, Donnita Tucker, and Wendy Johnson. 2021. Engaging Black Female Students in a Year-Long Preparatory Experience for AP CS Principles. *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* May 2020 (2021), 706–712. https://doi.org/10.1145/3408877.3432560

Paul Goldenberg, June Mark, Brian Harvey, Al Cuoco, and Mary Fries. 2020. Design principles behind beauty and joy of computing. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE* (2020), 220–226. https://doi.org/10.1145/3328778.3366794

Jeff Gray, Owen Astrachan, Kathy Haynie, Chinma Uche, Siobhan Cooney, Fran Trees, and Richard Kick. 2019. Infusing cooperative learning into AP computer science principles courses to promote engagement and diversity. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (2019), 1190–1196. https://doi.org/10.1145/3287324.3287421

References cont.

William Himmele and Pérsida Himmele. 2012. How to know what students know. *Educational Leadership 70*, 1 (2012), 58–62. https://pdo.ascd.org/LMSCourses/PD15OC007M/media/M2-TPTReading2.pdf

Andrew Hoegh and Barbara M. Moskal. 2009. Examining science and engineering students' attitudes toward computer science. Proceedings - Frontiers in Education Conference, FIE. https://doi.org/10.1109/FIE.2009.5350836

Beryl Hoffman, Ralph Morelli and Jennifer Rosato. 2019. Student Engagement is Key to Broadening Participation in CS. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. 1123–1129. https://doi.org/10.1145/3287324.3287438

Spencer Kagan and Miguel Kagan. 2009. Kagan Cooperative Learning. Kagan Publishing.

Spencer Kagan, Miguel Kagan, and Laurie Kagan. 2015. 59 Kagan Structures - Proven Engagement Structures. Kagan Publishing.

References cont.

Richard Kick and Frances P. Trees. 2015. AP CS Principles: Engaging, Challenging, and Rewarding. *ACM Inroads 6*, 1 (Feb 2015), 42–45. https://doi.org/10.1145/2710672

Clayton Lewis, Michele H. Jackson, and William M. Waite. 2010. Student and Faculty Attitudes and Beliefs about Computer Science. *Communications of the ACM 53*, 5 (May 2010), 78–85. https://doi.org/10.1145/1735223.1735244

June Mark and Kelsey Klein. 2019. Beauty and joy of computing: 2016-17 findings from an AP CS principles course. SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education (2019), 627–633. https://doi.org/10.1145/3287324.3287375

Melissa M. McDonald, Virgil Zeigler-Hill, Jennifer K. Vrabel, and Martha Escobar. 2019. A Single-Item Measure for Assessing STEM Identity. *Frontiers in Education* 4, July (2019), 1–15. https://doi.org/10.3389/feduc.2019.00078

References cont.

Linda J. Sax, Kaitlin N.S. Newhouse, Joanna Goode, Max Skorodinsky, Tomoko M. Nakajima, and Michelle Sendowski. 2020. Does ap cs principles broaden participation in computing? an analysis of apcsa and apcsp participants. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE* (2020), 542–548. https://doi.org/10.1145/3328778.3366826

Joshua Levi Weese and Russell Feldhausen. 2017. STEM Outreach: Assessing Computational Thinking and Problem Solving. In *2017 ASEE Annual Conference & Exposition*. ASEE Conferences, Columbus, Ohio. https://peer.asee.org/28845

David Weintrop and Uri Wilensky. 2017. Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education* 18, 1 (2017), 1–25. https://doi.org/10.1145/3089799

References cont.

Eric N. Wiebe, L. Williams, K. Yang, and C. Miller. 2003. Computer Science Attitude Survey. *Computer Science* January 2003 (2003), 5. arXiv:arXiv:1011.1669v3 http://www4.ncsu.edu/$\sim$wiebe/www/articles/prl-tr-2003-1.pdf

David Wolber. 2011. App Inventor and Real-World Motivation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) (SIGCSE '11). Association for Computing Machinery, New York, NY, USA, 601–606. https://doi.org/10.1145/1953163.1953329

Jeff Wyatt, Jing Feng, and Maureen Ewing. 2020. AP Computer Science Principles and the STEM and Computer Science Pipelines. December (2020). https://apcentral.collegeboard.org/pdf/ap-csp-and-stem-cs-pipelines.pdf?course=ap-computer-science-principles