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.)

  - 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?
<table>
<thead>
<tr>
<th>Approach</th>
<th>Exam Results</th>
<th>Self-efficacy and Confidence</th>
<th>Belongingness and Identity</th>
<th>Persistence and Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supporting Students through Peer Learning Communities [7]</td>
<td>✓</td>
<td>✓*</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Encouraging Cooperative Learning [9]</td>
<td>✓*</td>
<td>×</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Social Approaches</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finding the Beauty and Joy in Computing [8, 17]</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
<td>×</td>
</tr>
<tr>
<td>Engaging Students through Mobile Computing [5]</td>
<td>✓</td>
<td>N/A</td>
<td>N/A</td>
<td>✓*</td>
</tr>
<tr>
<td><strong>Curricular Approaches</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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&amp;course=bjc4nyc.html&amp;novideo&amp;noassignment


References cont.


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](https://doi.org/10.1109/FIE.2009.5350836)


References cont.


References cont.


References cont.


