Teaching Computer Science Principles
EDUC X300.44

5 Units
Summer 2017

Class Meeting Information

- Modality: Hybrid
- Online: June 26 - July 28 including weekly synchronous online classes:
  - Synchronous online classes will take place each week, Tuesdays at two times TBD (such as 4PM and 8PM)
  - Tool: Zoom
- Face-to-Face Fridays: June 30, July 7, July 14, July 21, July 28. 8am-12noon

Instructor Information

Name:	Beth Simon
Phone:	(858) 539-3247
Email:	bsimon@ucsd.edu
Website:	https://sites.google.com/ucsd.edu/bsimon sites.uci.edu/cs1c/teachingCSP

Note: general questions about the course (content questions, due date clarifications, etc.) should be posted via the general Q&A forum on our **course site** via the Learning Management System (LMS) so that all students can benefit from answers. To reach the instructor regarding personal questions (such as a request for an extension due to a family emergency or a request for an incomplete grade), please email cs1catoc@uci.edu

Dr. Beth Simon is an Associate Teaching Professor in the Department of Education Studies at UC San Diego. Her research interests lie in the areas of computing education and online and technology-enhanced teaching. Beth is currently involved in development of high school computing curriculum and the training and community needs of K-12 teachers wanting to bring computing education opportunities to their students. Previously, Beth has studied the impacts of evidence-based active learning practices (e.g., Peer Instruction) on student outcomes in higher education, student conceptions of computing concepts, and novice computing students' self-efficacy. Beth was one of the initial university pilot instructors for the AP CS Principles course. Beth currently serves as Faculty Advisor for Technology-Enhanced Teaching for UCSD's Teaching and Learning Commons, where she supports faculty and instructional staff in the use of technology to support their educational efforts both on-campus and through MOOCs (Massive Open Online Courses).

Beth formerly served as a Teaching Professor in the Computer Science and Engineering Department and as Director of UCSD's Center for Teaching Development (now part of the Teaching and Learning Commons). From 2014-2015, Beth served as the Principal Teaching and Learning Specialist at Coursera, supporting faculty in development of MOOCs and advising on pedagogical platform development. During 2007-2008, Beth served as a Science Teaching and Learning Fellow in the Carl Wieman Science Education Initiative at the University of British Columbia.

Course Description

Teaching Computer Science Principles (CSP) is a professional development course designed to support teachers in teaching the AP Computer Science Principles course or a similar computational thinking-based course. It is based on six *Computational Practices*: connecting computing, creating computational artifacts, abstracting, analyzing problems and artifacts, communicating, and collaborating. Additionally, the course focuses on seven *Big Ideas*: creativity, abstraction, data and information, algorithms, programming, the Internet, and global impact. This Teaching CSP course introduces computer science content through a variety of techniques including online videos (e.g. discussing student misconceptions, modeling classroom strategies, concept introductions, debugging advice examples, lesson plans overviews), guided engagement with lesson plans, scaffolded programming scrambles, online discussions, reflective writing, and guided lesson plan evaluation and peer review, in-person pair programming and peer instruction. While introducing the content and pedagogical knowledge, also focus on providing structure teachers can continue to use to build their computational thinking and programming skills as they teach the course. The course purports that CSP teachers and students are part of a learning community where all members contribute to the learning process in meaningful ways.

The Teaching CSP course and professional development consists of:

- A 5-week hybrid course (face-to-face and online) that develops teachers' pedagogical content knowledge and content knowledge for the AP CSP curriculum.
- A Professional Learning Community with workshops during the academic year to support teachers as they implement the curriculum for the first time.

Prerequisites — Classes or Knowledge Required Before Taking This Course

Teaching CSP is the second course in UCI's Computer Science Teacher Certificate Program, which is currently funded by the National Science Foundation. As such, it is expected that all participants will have completed the first course in the program, Teaching ECS. All students must be in-service teachers and admitted through that program.

Course Sequencing

The Computer Science Teacher Certificate Program consists of four sequential courses:

- Teaching Exploring Computer Science (hybrid F2F and on-line)
- Teaching Computer Science Principles (hybrid F2F and on-line)
- Advanced Topics for Computer Science Teachers (hybrid F2F and on-line)
- Computer Science Teaching Methods and Student Assessment (Face-to-Face)

Course Objectives

At the end of this course, students will be able to:

- Identify pedagogical content knowledge for teaching Computer Science Principles
- Elaborate on strategies of integrating computational thinking in teaching and learning
- Create an equitable computer science classroom environment
- Design effective instruction for Computer Science Principles
- Design student assessment and evaluation for the CS Principles Performance Tasks and written exam or similar activities
- Identify major components and functions of digital devices and computing systems???
- Identify the impacts of computing on society
- Develop computational artifacts using sequential execution, methods, parameters, events, mathematical expressions, functions, if statements, boolean expressions, loops and lists

Course Material

All course materials will be provided in the online LMS.

Course Outline

Week 1 Online (6/26/17)	**Topics/Objectives:** Peer Instruction, Creating a Supportive Classroom, Global Impacts, Programming Proficiency, Comparing Curriculum	**Key Topics:** ● Structures to engage students in deep learning ● Unconscious Bias ● Social, ethical and legal issues and impacts of computing ● Programming Concepts: sequential/parallel execution, methods, parameters **Learning Objectives:** ● Analyze the value of computational thinking for all students ● Review research on Peer Instruction and review materials for high school classrooms ● Discuss modifications of classroom practices for reducing unconscious bias ● Implement programs using methods, parameters, sequential execution, repetition ● Experience a scaffolded programming introduction ● Identify, summarize, and critique a TED talk on technology and society
	Learning Activities:	● Peer review assignment: Brainstorming access issues in your school (30 min) ● Programming assignments (10 hours) ● Peer Instruction videos and readings (1 hour) ● Classroom Resource Development (2 hours) ● Unconscious Bias videos and discussion (1 hour) ● Classroom Design Checklist (30 min) ● Exploration, summary and reflection, and discussion on technology and society (1.5 hours) ● Synchronous online class (1 hour)
	Assignments Due:	1. Brainstorming access list submission and review of three peers' submissions 2. Two online discussion forum posts and response to two peers' posts 3. Classroom Design Checklist 4. Five Alice programs

Week 1 Face-to-Face (6/30/17)	**Topics/Objectives:** AP CS Principles Curriculum Framework and Course Variants Peer Instruction, Creating a Supportive Classroom, Programming Proficiency	**Key Topics:** ● AP CS Principles Curriculum Framework and Course Variants ● Peer Instruction ● Unconscious Bias ● Programming Concepts: methods, parameters **Learning Objectives:** ● Experience Peer Instruction as a student ● Observe a teacher doing Peer Instruction ● Use Peer Instruction sentence starters ● Create a revised unconscious bias checklist ● Contrast programming instruction designs
	Learning Activities:	● Overview of AP CS Principles Framework and exploration of APCSP course variants (45 min) ● Peer Instruction student experience (30 min) ● Peer Instruction debrief (30 min) ● Unconscious Bias checklist Think/Pair/Share + update (45 min) ● Programming Concepts Q&A (30 min) ● Think/Pair/Share contrasting programming instruction designs (60 min)
	Assignments Due:	1. Revised Bias Checklist 2. Comparison/Contrast worksheet on Programming Instruction Design

Week 2 Online (7/3/17)	**Topics/Objectives:** Programming Proficiency, Explore Performance Task, Internet, Data	**Key Topics:** ● Internet, Data and Information ● The APCSP Performance Task Framework: Explore and Create ● Introduction to APCSP Explore Task ● Programming Concepts: events, mathematical expressions, functions, variables **Learning Objectives:** ● Explain how information is routed via the Internet ● Experience various types of lessons for teaching cybersecurity, encoding, and encryption ● Perform the APCSP Explore Task ● Apply the Explore Task Rubric to sample student work ● Create programs and functions which use basic Booleans expressions and mathematical expressions
	Learning Activities:	● Internet Online activities & reading (2 hours) ● Encryption activity review, critique and modification (2 hours) ● Read, perform, and submit APCSP Explore Performance Task (8 hours) ● Programming assignments (8 hours) ● Synchronous online class (1 hour)
	Assignments Due:	1. Two submissions to Internet activity lessons 2. Lesson modification and review of two peers' lessons 3. Submission of the APCSP Explore Task (written responses + computational artifact) 4. Three Alice programs

Week 2 Face-to-Face (7/7/17)	**Topics/Objectives:** Explore Performance Task, Developing Programming Proficiency	**Key Topics:** ● APCSP Performance Task: Explore ● Programming Concepts: events, mathematical expressions, functions, and variable **Learning Objectives:** ● Evaluate APCSP Explore Performance Task ● Deepen understanding of common student misconceptions around expressions, functions, and variables ● Critique a Peer Instruction session
	Learning Activities:	● Explore programming misconceptions through Peer Instruction (1 hour) ● Identify and evaluate resources for supporting Explore Performance Task (1 hour) ● Create personalized lesson Plan for Explore Performance Task (2 hours)
	Assignments Due:	1. Collaboratively produce list of resources for Explore Performance Task 2. Lesson plan for Explore Performance Task preparation

Week 3 Online (7/10/17)	**Topics/Objectives:** Pair Programming, Scaffolded Problem Solving, Developing Programming Proficiency, Curriculum Planning	**Key Topics:** ● Pair Programming and running a programming lab ● Scaffolded learning of problem solving ● Programming Concepts: if statements, compound Boolean expressions, nested if statements ● Curriculum planning ● Computer Science pencil & paper exams **Learning Objectives:** ● Plan and critique approaches for implementing Pair Programming and lab time ● Explore Analyze-Level options for developing programming knowledge and skills ● Create programs using if statements, nested if statements, and compound Boolean expressions ● Evaluate Course Planning & Pacing Guide ● Demonstrate to and guide students in using best practices for pencil & paper computing exams
	Learning Activities:	● Peer Review: APCSP Course Planning and Pacing Guide (3 hours) ● Videos, online discussion, and draft of plans for implementing Pair Programming (3 hours) ● Explore & reflect on programming scrambles to scaffold development of programming skills (2 hours) ● Create programs using if statements, nested if statements, and compound Boolean expressions (8 hours) ● Videos & peer review showing application of test taking recommended practices (1 hour) ● Synchronous online class (1 hour)
	Assignments Due:	1. Evaluation of the Course Planning and Pacing Guide and review of two peers' evaluations 2. Online discussion post and response to two peers' posts 3. Analysis of the Programming Scrambles experience and review of two peers' analyses 4. Two Alice programs 5. Submission of three exam questions using best test taking practices

Week 3 Face-to-Face (7/14/17)	**<u>Topics/Objectives:</u>** Pair Programming, Scaffolded Problem Solving, Developing Programming Proficiency	**Key Topics:** ● Pair Programming and running a programming lab ● Scaffolded learning of Problem Solving ● Programming Concepts: of Statements, compound Boolean expressions, nested if statements **Learning Objectives:** ● Experience Pair Programming ● Reflect on challenges of implementing Pair Programming ● Discuss differences in learning to program by creating vs. analyzing ● Deepen understanding of common student misconceptions around if statements, nested if statements and compound Boolean expressions
	Learning Activities:	● Experience Pair Programming and reflect (1 hour) ● Think/Pair/Share differences in programming by scramble and programming from scratch (40 min) ● Discuss assessment options using Programming Scrambles (20 min) ● Peer Instruction implementation practice and feedback (2 hours)
	Assignments Due:	1. Lesson plan for using Programming Scrambles 2. Teaching Demonstration: Peer Instruction

Week 4 Online (7/17/17)	**Topics/Objectives:** Computing Education Leadership, Create Performance Task, Developing Programming Proficiency	**Key Topics:** ● Developing a computing education culture in your schools and districts ● Introduction to APCSP Performance Task: Create ● Programming Concepts: loops, lists **Learning Objectives:** ● Explore approaches for leading computing education in your school ● Understand the APCSP Create Performance Task requirements ● Create programs using loops and lists ● Implement Curriculum Design for your class (e.g. Canvas or other)
	Learning Activities	● Reading and discussion of APCSP Create Performance Task Requirements (2 hours) ● Quiz of applying the APCSP Rubric (1 hour) ● Videos, readings, and web exploration for peer review of computing education leadership goals and concerns at your school (2 hours) ● Create programs using loops and lists (8 hours) ● Fill in first two months of curriculum platform for your class (4 hours) ● Synchronous online class (1 hour)
	Assignments Due	1. Online discussion post and response to two peers' posts 2. Online quiz applying the APCSP Create Rubric 3. Outline of leadership goals and concerns and review of two peers' outlines 4. Three Alice programs 5. PT Create Submission (1 video, 4 written responses)

Week 4 Face-to-Face (7/21/17)	**Topics/Objectives:** Computing Education Leadership, Create Performance Task, Developing Programming Proficiency, Curriculum Planning	**Key Topics:** ● Developing a computing education culture in your schools and districts ● APCSP Performance Task: Create ● Programming Concepts: loops, lists ● Curriculum planning **Learning Objectives:** ● Understand the various components of developing a culture of computing education in your environment ● Understand the requirements of the APCSP Create Performance Task and Evaluation Rubric ● Deepen understanding of common student misconceptions around loops and lists ● Reflect on challenges of curriculum planning
	Learning Activities:	● Structured sharing and discussion on computing education leadership (guest speaker) (1 hour) ● Question and answer period on APCSP Create Performance Task (30 min) ● Collaborative craft of Create Performance Task preparation activities (30 min) ● Peer Instruction implementation practice and feedback (2 hours)
	Assignments Due:	1. Collaboratively created leadership activity planner 2. Lesson plan draft for Create Performance Task preparation 3. Teaching Demonstration: Peer Instruction

Week 5 Online (7/24/17)	**Topics/Objectives:** Create Performance Task, Curriculum Planning, Exam Preparation Approaches	**Key Topics:** ● APCSP Create Performance Task ● Curriculum planning & pacing review ● Exam Techniques for Students **Learning Objectives:** ● Experience the APCSP Create Performance Task and submission process ● Implement curriculum design for your class (e.g. Canvas or other) ● Demonstrate to and guide students in using best practices for pencil & paper computing exams
	Learning Activities:	● Performing a guided practice for the APCSP Create Performance Task (Lyft) (6 hours) ● Videos & peer review showing application of test taking recommended practices (1 hour) ● Fill in scaffold materials for remaining months of curriculum platform for your class (6 hours) ● Synchronous online class (1 hour)
	Assignments Due:	1. Video demonstration (screen capture) of guiding students through a pencil and paper exam question 2. Identify and describe algorithms and abstraction from a program

Week 5 Face-to-Face (7/28/17)	**Topics/Objectives:** Create Performance Task, Curriculum Planning, Professional Learning Community	**Key Topics:** ● APCSP Create Performance Task ● Curriculum planning ● Professional Learning Community **Learning Objectives:** ● Prepare for supporting APCSP Create Performance Task ● Share planned practices from Curriculum Planning ● Prepare for PLC engagement throughout year
	Learning Activities:	● Discuss and co-edit APCSP Create Performance Task review, rubric revision, and student checklist/guide (1 hour) ● Peer instruction implementation practice and feedback (2 hours) ● Curriculum Planning share out (1 hour) ● PLC Walkthrough (1 hour)
	Assignments Due:	1. Teaching Demonstration: Peer Instruction

Evaluation and Grading

Evaluation of Student Performance Weighted as Percentages of the Total Grade

10%	F2F meetings attendance and participation
10%	Weekly synchronous online attendance and participation
30%	Computing artifacts and programming related quizzes
20%	Online material usage/completion
20%	Online engagement activities (discussion prompts, peer review assignments)
10%	Peer Instruction Sample Teaching Session

Categories of Points:

You will find that points for online activities fall into two general types: points for correctness and points for effortful participation. Points for correctness are given for activities where there are specific correct/incorrect answers (e.g. quizzes, prescribed programming assignments). However a lot of online activities (participation in discussion prompts, submitting peer review assignments and reviewing the work of others, contributing to googledoc-based resource creation, etc.) are graded for effortful participation. Effortful participation means you engaged with the activity in a meaningful way (e.g. not just responding "Great job!" in replying to a discussion response, completing peer reviews on time, and having roughly equal participation in contributing to googledoc resource creation (graphically viewable via the Chrome Add One Docuviz). Points for F2F meetings are given for effortful participation.

Grading Scale

Students have can choose to receive a letter grade or Pass/Not Pass.

If a letter grade is selected:

A = 90% – 100%
B = 80% – 89%
C = 70% – 79%
D = 60% – 69%
F = 59% or less

If P/NP is selected:

P >= 70%
NP < 70%

Code of Conduct

All participants in the course are bound by the University of California Code of Conduct, found at http://www.ucop.edu/ethics-compliance-audit-services/_files/stmt-stds-ethics.pdf

Netiquette

In an online course, the majority of our communication takes place in the course forums. However, when we have a need for communication that is private, whether personal, interpersonal, or professional, we will use individual email or telephone. Our primary means of communication is written. The written language has many advantages: more opportunity for reasoned thought, more ability to go in-depth, and more time to think through an issue before posting a comment. However, written communication also has certain disadvantages, such a lack of the face-to-face signaling that occurs through body language, intonation, pausing, facial expressions, and gestures. As a result, please be aware of the possibility of miscommunication and compose your comments in a positive, supportive, and constructive manner.

Academic Honesty Policy

The University is an institution of learning, research, and scholarship predicated on the existence of an environment of honesty and integrity. As members of the academic community, faculty, students, and administrative officials share responsibility for maintaining this environment. It is essential that all members of the academic community subscribe to the ideal of academic honesty and integrity and accept individual responsibility for their work. Academic dishonesty is unacceptable and will not be tolerated at the University of California, Irvine. Cheating, forgery, dishonest conduct, plagiarism, and collusion in dishonest activities erode the University's educational, research, and social roles.

Students who knowingly or intentionally conduct or help another student engage in dishonest conduct, acts of cheating, or plagiarism will be subject to disciplinary action at the discretion of UCI Division of Continuing Education.

Disability Services

If you need support or assistance because of a disability, you may be eligible for accommodations or services through the Disability Service Center at UC Irvine. Please contact the DSC directly at (949) 824-7494 or TDD (949) 824-6272. You can also visit the DSC's website: http://www.disability.uci.edu/. The DSC will work with your instructor to make any necessary accommodations. Please note that it is your responsibility to initiate this process with the DSC.