CSC 313: Teaching Computing <DRAFT 2/20 - expect revisions>

Professor: Dr. Zoë Wood and Dr. Bruce DeBruhl
Office: 14-209
Phone: (805) 756-5540
office hours: ——
lecture: ——
email: zwood@calpoly.edu

Course Description: *An introduction to pedagogical methods and practical techniques for computer science education: selecting appropriate content, designing assignments and activities, evaluating student learning, and evaluating teaching efficacy. Hands-on guided curricular design activities and real-world practice. 3 lectures, 1 laboratory. Prerequisite: CPE/CSC 202.*

Course Objectives: By the end of the quarter students will be able to:

- Explain common challenges to learning and teaching computing to broad audiences
- Discuss computational concepts and correct common misunderstandings
- Assess a proposed CS curriculum and teaching methodology on whether it is appropriate for a target audience
- Identify, evaluate, and effectively use evolving tools for computing education
- Identify, evaluate, and disseminate results from a research study
- Identify, evaluate, and disseminate changing pedagogical norms
- Design an effective assignment for teaching fundamental computational concepts

(Some of the) Computer Science and University learning objectives

1. An ability to apply knowledge of computing and mathematics appropriate to the discipline
2. An ability to analyze a problem and identify and define the computing requirements appropriate to its solution
3. An ability to function effectively on teams to accomplish a common goal
4. An understanding of professional, ethical, legal, security, and social issues and responsibilities
5. An ability to communicate effectively with a range of audiences
6. An ability to analyze the local and global impact of computing on individuals, organizations and society
7. Recognition of the need for, and an ability to engage in, continuing professional development
8. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
9. An ability to apply design and development principles in the construction of software systems of varying complexity

Assignments/Grade breakdown:

- 1 final assessment (24% of final grade)
- Weekly reading quiz and discussion participation (40% of final grade)
- ~10 Lab exercises (11% of final grade, 1-2% each)
- 2-3 large teaching computing related projects (tutoring center, K-12 outreach activity or library programming workshop) (10% each)

Final assessment: The final assessment for this course will consist of three parts: First, students will be observed and assessed during their tutoring center (or library workshop) practical experience. Second, the students will prepare a portfolio of educational materials gathered throughout the course. Lastly, students will prepare a reflection of their experience and changing perception in computing education.

Recommended Text: Readings related to computer science education (see schedule for example texts)

Class style and logistics
I expect you to participate in class and engage with the class material (studies suggest that taking longhand notes is one of the better ways to guarantee your engagement with the material in class)[1] I also expect you to form a community of scholars for the duration of the quarter (and hopefully longer). My teaching style is very interactive – if you want to know more about why see ([2]).

Laptops have been shown to be distracting in lecture[3] and are not allowed unless specified (or a specific exception is negotiated) -- same for cell phones.

Lecture and Lab Attendance: Attendance and participation in the course is mandatory. Participation includes responding to questions in class, lab or office hours and making observations or discussing course material in class, lab or office hours. Bi-weekly group exercises or pop quizzes may also required for participation.

Lab and Lab Exercises: Regular and frequent labs will be assigned and collected each week and, together with lecture attendance will comprise 16% of your course grade. The three hours of scheduled lab time each week is the primary time your instructor will be available for questions and assistance – ***make wise use of this resource!*** You are expected to work on the lab exercises during your scheduled lab time plus as much additional time as necessary to complete them. The lab exercises are designed to familiarize you

[1] http://www.theatlantic.com/technology/archive/2014/05/to-remember-a-lecture-better-take-notes-by-hand/361478/

[2] Applying the Seven Principles for Good Practice in Undergraduate Education" (1991) Chickering and Gamson

[3] http://www.yorku.ca/ncepeda/laptopFAQ.html

with some of the concepts necessary to complete your projects and to help you do well on quizzes and exams. You may work on your projects in lab ***after*** completing all currently assigned labs.
IMPORTANT: No late labs will be accepted.

Honesty: Although I encourage you to have lively discussions with one another, **all work you hand in must be your own work, unless otherwise specified**. If your program or parts of your program are plagiarized from another student or unapproved source, you will fail the course and a letter will be put in your file with Cal Poly Judicial Affairs.

Diversity and Inclusivity and Community: I consider this classroom to be a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities and expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the class. It is my goal that together we create an environment that facilitates and fosters access to computer science. We are all learners.

In practice in lab, I expect us to strive to build a community in which:

- We are not code snobs. We do not assume knowledge or imply there are things that somebody should know.
- After our work is complete, we prioritize the education of others and actively offer to help, explain, debug, etc. in order to support one another's learning. We do not share our working solution, but explain the logic/thinking behind our solution and help others recognize errors in their implementation when invited to do so.
- We consistently make the effort to actively recognize and validate multiple types of contributions to a positive classroom environment.

The following schedule for the lectures and assignments is tentative.

Week 0	4/2	Introduction: Why teaching programming is difficult
	Reading	“The buggy path to the development of programming expertise” (Pea, et. al)
Week 1	4/7	Theories of learning
	Reading	*“The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One tutoring”*
	4/9	Theories of learning
	Reading	About How the Mind Works and What It Means for the Classroom" by Daniel T. Willingham (Facts before skills)
Week 2	4/14	Survey of teaching pedagogy in CS “pair programing peer instruction, PBL) effective presentations
	Reading	A Comparison of Lecture-based and Active Learning Design Patterns in CS Education
	4/16	Survey of teaching pedagogy in CS “pair programing peer instruction, PBL) effective presentations
		a chapter from The Cambridge Handbook of Computing Education Research, 2019, written by Kathi & Shriram. https://cs.brown.edu/~sk/Publications/Papers/Published/kf-prog-paradigms-and-beyond/paper.pdf
Week 3	4/21	Choosing and designing content for your audience with a focus on course learning objectives
	Reading	***"A theory of instruction for introductory programming skills".*** *Computer Science Education*
	4/23	Choosing content for inclusion and diversity
	Reading	"The anatomy of interest Women in undergraduate computer science" Jane Margolis
Week 4	4/28	Choosing content for inclusion and diversity
	Reading	Race and gender differences in how sense of belonging influences decisions to major in STEM, Katherine Rainey , Melissa Dancy, Roslyn Mickelson, Elizabeth Stearns and Stephanie Moller
	4/30	Designing and iterating on computing assignments
	Reading	
Week 5	5/5	Assessing student learning and evaluations including rubrics

	Reading	“Are automated assessment tools helpful in programming courses?”
	5/7	Assessing student learning and evaluations including rubrics
	Reading	"Assessing Incremental Testing Practices and Their Impact on Project Outcomes"
Week 6	5/12	Evaluating your efficacy and teaching
	Reading	“Speech and drama training for lectures as a means of improving university teaching ”
	5/14	CS education research methods and ethics
	Reading	"Whose culture has capital? A critical race theory discussion of community cultural wealth" by Tara J. Yosso
Week 7	5/19	Mastery
	Reading	“Grading for Enhanced Motivation and Learning”
	5/21	CS education research methods and ethics
Week 8	5/26	The culture of computer science education
	Reading	"Culture of Disengagement in Engineering Education?"
	5/28	Student mental health in CENG
		"Characterizing Mental Health and Wellness in Students Across EngineeringDisciplines" (Danowitz et al)
Week 9	6/2	Current topics
	Reading	*Depth first search*
	6/4	Current topics
Final		

——-

Tutoring Center shadow and lab session or K12 outreach plan:

In second week of the quarter:

- Every student in 313 will sign up to shadow an existing tutor for 1 hour in one of the 5 nights of tutoring (two 313 students per tutor = 4 people a night over a 2 week period, with 2 tutors being shadowed).
- Then ~20 people from the class will sign up for 2 hours of tutoring (1 person per night) <over the 5 nights of tutoring = week 3-7 of the quarter>.
- Those who do not sign up for a night of tutoring must do a K12 outreach event.