Course Code: Computer Science Teaching – MAED 598
Semester Year: Spring 2022
Credit Hours: 3 credits
Class Meeting Times: T/Th 7:00 – 8:15 pm
Class Location: MZ 022

Instructor: Christine Liebe
Office Phone: 970-274-6688
Office Hours: by appointment

Email: cliebe@mines.edu
Office Location: CTLM 262
Instructor Website: https://christineliebe.com/

Pre-requisites: CSCI 101, CSCI 261

Required Technology: Canvas, laptop or tablet

Course Description:

In MAED 435 students will learn about CS curriculum, instruction, and assessment by actually teaching from elementary, middle school, and high school lesson plans. Through deconstructing practice teaching experiences, students will explore classroom connections to theory, instructional strategies, classroom management, and CS content. Students will develop weekly lesson plans, observe one K-12 CS classroom, learn from teacher guest speakers, and develop a teaching module. Through experience, reflection, observation, writing, study, and discussion, students will construct knowledge (social constructivism). By producing lesson plans and modules, students will construct artefacts of knowledge, portfolio pieces (social constructionism).

Learning Outcomes:

At the completion of the course, you will be able to:
Course level learning outcomes go here.

1. Nurture development of computer science (CS) processes and practices. They anticipate how students' use of CS practices will look within specific grade-band topics.
2. Identify, adapt, or develop lessons that explicitly teach computational thinking and CS processes and practices demonstrating these as tools use to solve problems and communicate ideas.
3. Demonstrate that doing computer science is a creative process that promotes critical thinking, perseverance, problem posing, and problem solving through iterative testing.
4. Provide examples and connections for students to see that computational thinking is a human endeavor that is practiced in and out of school, across many facets of life.
5. Integrate the history of CS into content, including contributions from people with different gender and cultural, linguistic, religious, and racial/ethnic backgrounds.
6. Articulate how CS is based on mathematics and constructed conventions and agreements about the meanings of words and symbols, which vary depending on

the computer language.

7. Cultivate their students' CS identity by helping students realize the usefulness of CS by providing connections to students' everyday lives and building their students' CS self-efficacy by encouraging persistence from every student and demonstrating the belief that every student is capable of learning and expressing their creativity and intelligence with CS.
8. Identify and implement practices that draw on students' previous background experience with CS, cultural, and linguistic resources/strengths and challenge practices grounded in deficit-based thinking.
9. Select, adapt, or develop lessons that explicitly engage students in the CS practices defined in the Computer Science Teachers Association and in the Colorado High School Computer Science standards.
10. Select and develop appropriate formative and summative CS assessments including student portfolios, websites, and online repositories.
11. Encourage diverse secondary students into their CS courses and support/encourage them using best practices known.
12. Identify, adapt, or develop lessons that reflect the interconnectedness of content areas/disciplines to help erase the disciplinary lines and reflect authentic situations.
13. Create a mini-unit or module (3 days or more) that explicitly teaches some aspect of CS practices or the social/ethical context of CS.
14. Clearly articulate their CS ideas in writing. Analyze text based on occasion, audience, form and function. Compose one page reflections with an awareness about introductions, conclusions, and topic sentences. Articulate the process of and compose with an awareness about the composing process which is an iterative process of formulation, composition, and revision. Incorporate and cite correctly all evidence used to support a text's claim/s.
15. Clearly articulate their CS ideas verbally. Delineate effective characteristics of multi-media presentations. Articulate CS practices in a way that secondary students can understand and be motivated to explore these practices. Collaborate with others towards giving and receiving feedback on both oral and written work about teaching CS as a community of inquiry.

Learning Environment:

Fundamentally, we expect and require respect in this course for yourself practicing teaching professionalism, your classmates, and the instructor.

- Respect for yourself includes taking care of yourself physically and mentally and advocating for an environment that facilitates learning for you.
- Respect for your classmates includes recognizing and appreciating the diversity of backgrounds and experiences of your classmates and making it your interest to foster a learning environment for everyone; all are welcome.
- Respect for the instructor (as well as your classmates) includes not participating in disruptive or distracting behavior: talking, playing games, or web surfing during lecture, for instance, make it difficult for others to focus on the reason we are all here.
- Respect must be mutual to be effective; the instructor will be held to the same standards of respect.

Required Text

Computer Science in K-12: An A-Z Handbook on Teaching Programming, edited by Shuchi Grover. (**ISBN-13:** 978-1734662702; **ISBN-10:** 1734662700)

Course Schedule

The syllabus and course schedule are subject to change throughout the semester. Students will be notified of any changes.

Week	Concepts	Practices	Theory	Text-book	Assignment
1	Algorithms	Digital Citizenship, Cyber hygiene	Knowledge, Skills, Attitudes, Beliefs	Ch. 1, 2, 3	LP
2	Data Structures	Before you program plan	Computational thinking	Ch. 4, 5	
3	Intro Programming Languages	Reading + Writing (Xie)	Deeper Learning, Interdisciplinary CS	Ch. 9, 10, 11	
4	Variables	Testing and Debugging	Student Agency	Ch. 8, 22	LP Analysis - 1
5	Operators and Expressions	Good habits of Programming	Cultural Relevancy	Ch. 12, 15, 25	
6	Physical Computing	Classroom management	SAMR	Ch. 6, 16, 24	O/LP, reflection
7	Selection with Conditionals	Cup method	TPACK	Ch. 7, 19	practice midterm
			MIDTERM		MIDTERM
8	Events	Modeling	TripleE	Ch. 14, 17	O/LPF, Present, reflection
9	Modularity	Incremental testing + development	Critical Thinking + Assessment	Ch. 13, 20	O/LPF, Present, reflection LP Analysis - 2
10	Recursion	Abstraction	Differentiation, CS-SCIC	Ch. 21	O/LPF, Present, reflection
11	Apps	GUI best practices	Spiraling Curriculum	Ch. 18, 23	O/LPF, Present, reflection
12	Hardware	Imagination	Self-Efficacy,Motivation, Perseverance	Ch. 26	Module – 1 O/LPF, Present, reflection
13	Networks	Electricity, binary	Identity, Belonging		Module – 2 O/LPF, Present, reflection
14	AP Exams	Test prep strategies	Learning Diversity		Module – 3 O/LPF, Present, reflection
15	Review	role play - jigsaw	Ethics and Development		Module Final, O/LPF, Present, reflection practice final
16			FINAL		FINAL