MSU Denver

CS 1030: Living in a Computing World

Syllabus (aka Course Policies)

Instructor: Ivo Georgiev, PhD
E-mail: igeorgi1@msudenver.edu
Cell: 415-297-8765

Time: MW, 10:00-11:50
Duration: 16 weeks (including Finals Week)
Location: Central classroom Room 101 (CN-101)
Office hours: On Slack (https://cs1030f16.slack.com/messages/general/) by appt

Catalog description (catalog entry): This course provides fundamentals needed to effectively cope with, understand, and be successful in a world with pervasive computing. The coursework explores impacts of computing (from social, ethical, economic, technical, legal, philosophical, and cognitive perspectives) and the related aspects of creativity, innovation, problem solving, critical thinking, collaborative teamwork, and multi-modal communication. Students participate in active-learning experiences and create materials using iterative processes similar to those used by artists, musicians, and engineers.

Required reading (book page): H.Walker, “The Tao of Computing”, 2nd ed, CRC Press

Course materials:

1. Latest syllabus, quizzes, exams, slides, activities are Moodle (enrollment key *cs1030f16*) at https://gouda.msudenver.edu/moodle/course/view.php?id=68
2. Backup for course materials, programming assignments are on Github (public repository) at https://github.com/ivogeorg/cs1030
3. SNAP programming assignments are online through any modern Web browser (HTML5 and JavaScript support required) at http://snap.berkeley.edu/ (instructions on Moodle)
4. For collaboration, online office hours: Slack

Recommended external materials:

1. (edX MOOC page) The Beauty and Joy of Computing (FREE)

2. (Coursera MOOC page) Learning How to Learn (FREE)

Grading weights:

1. 50% Online assignments (5 or 6)
2. 25% Online exams (Midterm and Final)
3. 15% Online quizzes (15 or 16)
4. 10% Class activities (29 or 30)

Grading policies:

1. All quizzes and exams will be on Moodle. Each quiz/exam will be available for a window of time, in which it can be taken. Each will be timed and there will be only one attempt. E.g. A quiz is available for a certain week and it allows 90 minutes to take. You can start it at any time during the week and have to finish it before 90 minutes. If you don't finish it, whatever portion you have finished will be counted toward your submission. No restarts are guaranteed.
2. Absolutely no late assignments. No partial credit after the due date. For quizzes and exams this is automatic. For programming assignments it will be also strictly enforced.
3. Attendance will be taken. You can miss 5 lecture periods for whatever reason w/ or w/o notice.
4. Students have to form teams of 2-4 people.
5. Quizzes and exams are individual submissions. Programming assignments are team submissions. Teammates get their team's grade for team submissions.

Grading scale:

Letter	**Low %**	**High %**
A	94	100
A-	90.60	<94.00
B+	87.20	<90.60
B	83.80	<87.20
B-	80.40	<83.80
C+	77.00	<80.40
C	73.60	<77.00
C-	70.20	<73.60

D+	66.80	<70.20
D	63.40	<66.80
D-	60.00	<63.40
F	00.00	<60.00

Withdrawal dates:

- Sun, Aug 28: 100% refund
- Wed, Sep 7: 50% refund
- Wed, Nov 9: Last day for W

Holidays:

- Sep 5: Labor Day
- Nov 21-27: Fall Break

Course schedule:

#	Week	Mon	Wed	Quiz/exam	PA
1	Aug 22	Overview	Tao1, Tao2	-	-
2	Aug 29	Tao1, Tao2	Tao2, Tao3	-	-
3	Sep 5	**Labor Day**	Tao4, Tao5	Q1	-
4	Sep 12	Tao4	-	Q2	Snap1
5	Sep 19	Tao6	Tao7	-	Snap1
6	Sep 26	Tao8	Tao9	Q3, Q4	Snap1
7	Oct 3	Tao9, Tao10	Tao11	Q5, Q6	Snap2
8	Oct 10	Tao12	-	Midterm	Snap2
9	Oct 17	Tao13	RoR	Q7, Q8	RoR1
10	Oct 24	-	-	Q9, Q10	RoR1
11	Oct 31	-	-	Q11, Q12	RoR1
12	Nov 7	Tao15	Tao16	Q13, Q14	RoR1

13	Nov 14	Tao17	-	Q15, Q16	RoR2
14	Nov 21	**Fall Break**	**Fall Break**	-	-
15	Nov 28	Ethics	-	Q17	RoR2
16	Dec 5	-	-	-	RoR2
17	Dec 12	**Finals Week**	**Finals Week**	Final	-

General departmental policies:
https://mcs.msudenver.edu/degrees/courses/policies

Fundamentals of the Computer Science Principles curriculum framework:
Course content integrates *big ideas in computing* with *computational thinking practices,* both outlined below, such that all practices and big ideas are represented. [Reference: The College Board. (2016) *AP Computer Science Principles: Course and Exam Description.* New York, College Board.]

I. Big Ideas in Computing
   A. **Creativity**: Computing is a creative human activity
      1. Creative development can be an essential process for creating computational artifacts
      2. Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem
      3. Computing can extend traditional forms of human expression and experience

   B. **Abstraction**: Abstraction reduces information and detail to facilitate focus on relevant concepts
      1. A variety of abstractions built on binary sequences can be used to represent all digital data
      2. Multiple levels of abstraction are used to write programs or create other computational artifacts
      3. Models and simulations use abstraction to generate new understanding and knowledge

   C. **Data and Information**: Data and information facilitate the creation of knowledge
      1. 1. People use computer programs to process information and to gain insight and knowledge

2. Computing facilitates exploration and the discovery of connections in information
3. There are trade-offs when representing information as digital data

D. **Algorithms**: Algorithms are used to develop and express solutions to computational problems
    1. Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages
    2. Algorithms can solve many, but not all, computational problems

E. **Programming**: Programming enables problem-solving, human expression, and creation of knowledge
    1. Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society)
    2. People write programs to execute algorithms
    3. Programming is facilitated by appropriate abstractions
    4. Programs are developed, maintained, and used by people for different purposes
    5. Programming uses mathematical and logical concepts

F. **The Internet**: The Internet pervades modern computing
    1. The Internet is a network of autonomous systems
    2. Characteristics of the Internet influence the systems built on it
    3. Cybersecurity is an important concern for the Internet and systems built on it

G. **Global Impact**: Computing has global impact
    1. Computing enhances communication, interaction, and cognition
    2. Computing enables innovation in nearly every field
    3. Computing has global effects — both beneficial and harmful — on people and society
    4. Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used
    5. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources

II. Computational Thinking Practices

A. Connecting Computing
    1. Identify impacts of computing
    2. Describe connections between people and computing
    3. Explain connections between computing concepts

B. Creating Computational Artifacts

1. Create a computational artifact with a practical, personal, or societal intent
2. Select appropriate techniques to develop a computational artifact
3. Use appropriate algorithmic and information management principles

C. Abstracting

1. Explain how data, information, and knowledge are represented for computational use
2. Explain how abstractions are used in computation or modeling
3. Identify abstractions
4. Describe modeling in a computational context

D. Analyzing Problems and Artifacts

1. Evaluate a proposed solution to a problem
2. Locate and correct errors
3. Explain how and artifact functions
4. Justify appropriateness and correctness of a solution, model, or artifact

E. Communicating

1. Explain the meaning of a result in context
2. Describe computation with accurate and precise language, notations, or visualizations
3. Summarize the purpose of a computational artifact

F. Collaborating

1. Collaborate with others in solving a computational problem
2. Collaborate with others in producing an artifact
3. Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of team members
4. Exchange knowledge and feedback with team members
5. Review and revise collective work as needed to create a high-quality artifact