INFO 1201 Computational Reasoning

Summer 2018

Instructional Team

Instructor

Wendy Norris, PhD Student, Information Science
wendy.norris@colorado.edu

Peer Assistant
TBD

Meeting Times

Class: Monday - Friday 11am-12:35pm
Location: KTCH 1B71 || Map: https://www.colorado.edu/map/?id=336&mrkId=193861

Office Hours: Wed and Fri 12:35 pm
Location: KTCH 1B71

Overview

Computing and information technologies permeate all aspects of our lives. They inspire how we connect with each other online through social networks and how we find information through search engines. Technologies also drive our physical world in how we navigate transportation systems and how we manage money on banking applications. Everyone should have the ability to not only use and interact with computing, but to also create and express themselves with computing.

This course is a hands-on introduction to create, invent, and build with computer programming. No programming experience is necessary and all backgrounds are welcome. Students will become exposed to high-level computational concepts and practices that include algorithms, data, parallelism, abstraction, and debugging. Assignments and projects will involve learning to program using the Scratch and Python programming languages. The creative and problem-solving strategies introduced in this course are applicable across many domains beyond information and computer sciences.

Goals

- Develop Computational Thinking concepts, practices, and perspectives to create, explore, and understand the world in new ways. In this class, you will engage with computational thinking through creative expression with computer programming. These concepts, practices, and perspectives are applicable beyond this class into other domains and interests beyond computing.
- Create and share interactive media projects, such as animations, games, and stories with Scratch and Python
- Collect, examine, and interpret large datasets with Python
- Connect the ways that computing interacts with many part of our lives
- Learn about opportunities that exist to extend and expand on the practices and activities in this class within and beyond CU

Course Format

The summer version of INFO 1201 consists of four classes and one lab each week. There are no planned exams or quizzes. Nor is there an assigned textbook. All course materials and resources will be in a universally accessible format and available on Canvas.

Class time is interactive and participatory, with students engaging in hands-on activities, small-group challenges, and showcases to share projects with their fellow students. Lab time is devoted to working on projects, reading assignments, and written reflections.

The 5-week summer class is divided into 6 modules that encapsulate a set of computational thinking topics, activities that develop practice and understanding of those topics, and projects that students will be responsible for creating. The instructor will also share real-world examples from journalism, strategic communication, politics, culture, etc., that connect to the topics presented during that module.

This format is meant to support a class structure where we can explore these ideas together, work on challenges together, learn from one another, and share our process of figuring things out (which will include lots of mistakes and detours) in addition to sharing our projects.

Programming Languages

This course will introduce students to two free programming languages:

Scratch
Scratch (scratch.mit.edu) is a programming language and an online community that allows people to create and share interactive media, such as animations, games, and stories. It was especially designed for children, ages 8 and up, but Scratch is being used by people of all ages around the world and millions of projects have been shared in its dynamic online community. It is a real programming language and the thinking and practices that you engage in as you create projects will be applicable to other programming, design, and creative endeavors. Scratch is free and can be found at: http://scratch.mit.edu

Python 3.6
Python is a programming language supported by a vast passionate community of programmers and has become an essential tool for many engineers, scientists, and researchers from a variety of fields that include astronomy, journalism, social science, and web development. There are many free resources and open communities that support people to create, build, and invent with Python. We'll be using Python version 3.6 with the free development environment called Jupyter Notebook and the Anaconda 3 distribution of python and libraries. ***Please note: There are different versions of Python. Your projects must be completed in Python 3.6+ or they will not be accepted. DO NOT USE PYTHON 2.***

Expectations

Culture

Everyone who has ever learned to program knows how it can be both hard and fun. It's hard because it feels like learning a foreign language — it's confusing and frustrating not knowing so many things right away, but through practice and time you'll learn to express yourself fluently. It's fun because you'll get to do exciting things with computing that you may have never done before and this is just the beginning. To help people learn through this hard and fun process, we want to create a culture where people will feel welcome to take creative risks, to make mistakes, and to celebrate the progress we make. To this end, the course instructional team will do their best to ensure equitable and inclusive participation in this class. We ask that you also create and promote this environment with us.

Collaboration

We encourage you to support and learn from one another throughout the course. In this course, collaboration means helping each other by troubleshooting someone's else project code, talking through a new concept in order to understand it, or sharing an idea to fulfil a required project component, etc. It does NOT mean group projects.

When working on course materials and projects together, please list your collaborators and describe how you worked together (e.g., if you looked up information online or how you and your peers came up with the solution). Whether or not you collaborate, you are each responsible for learning and understanding the topics and ideas in this course. **Each student is responsible for submitting his/her own assignments.** ***Copying code or otherwise submitting work that is not your own is considered an Honor Code violation and will be taken seriously.***

Technology

Laptops and other devices
This is an interactive, participatory, and hands-on class. ***Please bring your laptops to lecture and lab.*** If you do not own a laptop, or if bringing a laptop to class is an issue, please speak with your instructor and we will come up with a solution.

No cell phones for non-class purposes: No talking on your phone. No checking your email on your phone. No texting on your phone. They are welcome as data and creative tools in our classroom unless they become a distraction.

Use laptops for class purposes only: All the same rules for your phone apply to your computer.

Communication

We encourage you to ask questions and share reflections and feedback about the course. There are multiple ways to reach your peers and instructors for help and questions:

Canvas discussion forums: INFO 1201 uses Canvas as a means to ask questions, get clarification about an assignment, debug your code, etc. **Plus, you earn participation points (see: Participation and Attendance below) for asking or responding to others questions.** Instructors and peer assistants will check the forums regularly and aim for a response within 24 hours.

Office Hours: Instructors and peer assistants will host 2 office hours during the week paired with new lecture material and assignment deadlines. We welcome you to attend.

Email: The instructor is accessible by email **for any private questions/needs about the course. However, if you need help with a class assignment, please use Canvas to reach me. I will not respond to assignment questions via email.**

Late Work and Extensions

The summer course has been structured to avoid weekend homework/assignments, whenever possible. However, students should commit time M-F to mastering the topics, practicing, and reading the course materials/resources.

We can't emphasize enough the importance of keeping up with the pace of the class and completing activities and projects on time. This class moves very quickly and if you fall behind on the assignments and activities it will be very hard to catch up.

All assignments are due in Canvas by 11:59 p.m.

Assignments submitted after the deadline will not be graded and will earn a zero.

No extensions will be granted without a compelling reason due to circumstances beyond your control and which must be documented by a healthcare provider letter, military activation order, obituary/memorial service notice, police/fire report, etc.

If there is a campus emergency (e.g., weather, closure, etc.), the instructor will send a message to the class with directions about assignments, deadlines, etc.

Participation and Attendance

Participation is an essential part of how you will learn in this class. **It will be worth 20% of your grade.** By actively participating in the class, you can develop your skills and understanding of course concepts. All students are expected to come to class having made progress with course assignments and projects and fully participate fully in class activities and discussions.

Here are some examples of how students can earn participation points:

- Practice active listening – be attentive and be engaged
- Ask questions – there are no dumb questions; also don't be afraid to politely challenge
- Comment, build on, or clarify others' contributions
- Submit a thoughtful response to the Daily Note
- Post useful or interesting information to the Canvas class discussion forums
- Visit the instructors or teaching assistant to chat, ask questions, or give feedback
- Engage in and contribute to small group activities such as the debug challenges
- Document your participation by uploading a file to the Daily Note
- Give feedback and ask questions when someone shares their project

We understand that emergencies and disruptions happen, so if you have to miss a class session, we ask that you consult the syllabus, Canvas, and/or your peers to learn about what happened during class. It is still your responsibility to submit assignments and projects on time.

Assignments and Projects

Projects - *due at 11:59 p.m., days vary*

With every module, students will have an opportunity to create a project that builds on the topics introduced during that module. Each project will have some core criteria that students must be able to develop, but they will also have the opportunity to express their ideas and take the project in their own directions. Details can be found on Canvas.

Assigned projects for Summer Session A:	Mod 1 Scratch: Make It Fly
	Mod 2 Python: Mad Libs
	Mod 3 Python: Choose Your Own Adventure game
	Mod 4 Python: Exploring Data
	Mod 5 Python: Visualizing Data
	Mod 6 Python: Telling Stories

Students will submit ***2 project files to Canvas for each project***: One .ipynb (Jupyter Notebook) and one HTML file. The Canvas assignment tab will include step-by-step instructions on how to download/upload these files.

Project Reflections - *due at 11:59 p.m., days vary; in conjunction with projects*

In addition to the projects, students will also submit very brief reflections via a Google Form describing: 1) something that was fun about the project, 2) something that was hard and how they addressed it, 3) something that they might do next if they had more time, and 4) a brief explanation of the computational thinking concepts/practices/perspectives used in the project.

Project Showcase - *the next class day after the project/reflection due dates*

Each student will showcase twice over the course of the summer: 1) Choose ***one of the Mod 1-5 projects*** + 2) ***Module 6 Superhero Data Narrative (all students will showcase this project).*** Showcase is a 2 minute, in-class demo that describes the computational thinking concepts, practices, and perspectives used to create the project. Students can pre-request a showcase slot on the project reflection form. Otherwise, students will be called upon in class.

Reading Responses - *due at 11:59 p.m., days vary*

In each module, we'll curate a set of brief readings and/or podcasts that will further elaborate the topics of the module and help connect the module to real-world examples. Students will reflect on the readings by sharing brief responses to reading prompts via a Canvas quiz. In your responses, make sure to write clearly and concisely, to discuss the readings in your own words, and to make connections to what we've done in class or to your other classes and interests.

Final Course Reflection Paper - *due no later than 11:59 p.m. on Friday, July 6*

The course final is a reflection paper (minimum 750-words) that connects the computational thinking concepts, practices and perspectives to the various projects/activities accomplished over the course of the summer. Specific writing prompts will be provided ahead of the due date. Pro tip: Thoughtful, descriptive weekly project reflections (see project section above) can help scaffold a strong foundation for the final paper.

Assessment and Grading Rubric

Detailed rubrics will be distributed for each assignment. In this course, grading begins at zero and works up by accumulating points. Overall, the course grade will be determined by 4 criteria:

1. Projects/Reflections/Showcase combined (50% total grade)
    6 projects* = 10 pt scale = 60 pts total
    6 reflections = 4 pt scale = 24 pts total
    2 showcases = 8 pt scale = 16 pts total

2. Final Paper (20% total grade)
    1 paper = 100 pt scale = 100 pts total

3. Class participation (20% total grade)
    5 weeks = 20 pt scale* = 100 pts total

4. Reading Responses (10% total grade)
    10 reading responses* = 10 pt scale = 100 pts total

*extra credit is possible for outstanding work that exceeds the rubric criteria

The course will use a standard 100-pt scale to calculate the final grade:

Percentage earned	Grade	Percentage earned	Grade	Percentage earned	Grade
		93.0% and above:	A	90.0%–92.9%:	A-
87.0%–89.9%:	B+	83.0%–86.9%:	B	80.0%–82.9%:	B-
77.0%–79.9%:	C+	73.0%–76.9%:	C	70.0%–72.9%:	C-
67.0%–69.9%:	D+	63.0%–66.9%:	D	60.0%–62.9%:	D-
Below 60.0%:	F				

IMPORTANT: This course does not curve grades. The individual grades earned by a student for each the 4 criteria (listed above) will comprise the final grade. Each project handout will include detailed description of the grading rubric and point scales. Please read each handout thoroughly so there are no surprises.

Schedule Details

Part 1: Creating and Expressing with Scratch

Module 0: Computational Concepts + Practices

In the last decade, computational thinking has been recognized as an important of computational concepts, practices, and perspectives to help people understand and see our digital and networked world in new ways. This class focuses on developing computational thinking through creative expression with computer programming. Our first module will start our exploration of computational concepts, such as sequences, events and loops, using the Scratch programming language.

Class Sessions

Date	Topics	Things to turn in today
Mon 6/4	What is Computational Thinking (CT)? Introducktions Class culture <> CT practices Computer basics (directories, file naming) Syllabus, Canvas and questions/concerns	1. Reading response 0 2. Daily note
Tue 6/5	Intro to Scratch CT: Sequence, events and loops	1. Reading response 1 2. Daily note

Module 1: More Computational Concepts + Practices

We'll continue our exploration of computational concepts in this module as well as expand on computational practices. These practices are cultivated as you engage in the design and problem-solving process of creating and building your project.

Date	Topics	Things to turn in today
Wed 6/6	CT: Loops and conditionals Intro Make it Fly (MIF) project Work on Make It Fly (MIF) project	1. Reading response 2 2. Daily note
Thur 6/7	CT: Parallelism, operators and data Work on Make It Fly (MIF) project	1. Daily note
Fri 6/8 Lab	CT Concepts and Practices refresh Debug Challenge Work on Make It Fly (MIF) project	1. MIF project 2. MIF reflection 3. Daily note

Part 2: Playing and Experimenting with Python

Module 2: Computational Concepts, Practices + Perspectives // Python Basics: Variables, Datatypes, String Manipulation

In creating, designing, and building with computing, you can develop new perspectives about how you see yourself and how you see the world around you. We'll discuss computational perspectives that can emerge in this module.

We'll transition to Python in this next series of modules with the first set of modules focusing on the basics of Python. This module includes datatypes such as strings and integers and using variables in Python. While the language and the programming environment are new, the computational concepts, practices, and perspectives that you can engage in cut across all programming languages.

Date	Topics	Things to turn in today
Mon 6/11	Make It Fly (MIF) Showcase CT: Express, connect, question and intention Scratch to Python transition Jupyter Notebook and Anaconda install	1. Reading response 3 2. Daily note
Tue 6/12	Intro to Python and Jupyter Python basics 1: Datatypes, string manipulation, variables, and inputs Intro Mad Libs (ML) project	1. Daily note
Wed 6/13	CT concepts, practices and perspectives refresh Python basics 1 practice Work on Mad Libs (ML) project	1. ML project 2. ML reflection 3. Daily note

Module 3: Python Basics: Boolean, Control Flow, and Loops

We'll continue our dive into Python basics exploring boolean operations, control flow, and loops. These topics will allow us to control how our program makes decisions about what to do. These programming constructs are not so different from the Scratch programming blocks "if" and "repeat" or "forever".

Date	Topics	Things to turn in today
Thur 6/14	Mad Libs (ML) Showcase Python basics 1 refresh Python basics 2: Boolean, control flow and loops Intro Choose Your Own Adventure (CYOA) project	1. Reading response 4 2. Daily note
Fri	Python basics 1 and 2 practice	1. Daily note

6/15 Lab	Debug Challenge Work on Choose Your Own Adventure (CYOA) project	
Mon 6/18	Python basics 3: More loops and functions CT concepts refresh Work on Choose Your Own Adventure (CYOA) project	1. CYOA project 2. CYOA reflection 3. Daily note

Module 4: Python Basics: Creating Functions, Parameters, and Return Values // Exploring Data

Functions contain a block of code that can be reused over and over again. This ability can help simplify and manage a complex program. We'll learn to create functions and how to take advantage of their input (parameters) and output (return) capabilities.

Building on our practice from the past modules, we'll dive into something that Python is really great at: exploring data and lots of it. We'll learn about lists to help us manage and manipulate sets of data. Finally, we'll introduce you to taking advantage of Python libraries, or collections of functions that other Python developers have created for you to reuse.

Date	**Topics**	**Things to turn in today**
Tues 6/19	Choose Your Own Adventure (CYOA) showcase Python basics 4: More functions, parameters and return values Python basics 1-2-3-4 practice CT concepts, practices and perspectives refresh Pulling simple insights from data Intro Exploring Data (ED) project	1. Reading response 5 2. Daily note
Wed 6/20	Exploring Data: csv files, csv python module, and importing csv files Python basics 5: Strings and simple lists Python basics 1-2-3-4-5 practice Pulling simple insights from data, cont. Work on Exploring Data (ED) project	1. Reading response 6 2. Daily note
Thur 6/21	Python basics 6: Lists, tuples and dictionaries Python basics 1-2-3-4-5-6 practice Work on Exploring Data (ED) project	1. Daily note
Fri 6/22 Lab	Python basics 1-2-3-4-5-6 practice Debug Challenge Work on Exploring Data (ED) project	1. ED project 2. ED reflection 3. Daily note

Part 3: Computing and Society

Module 5: Visualizing Data

We can use storytelling techniques to help people understand insights from the data. We'll learn to use different python libraries to help us create visualizations of the Marvel and DC data.

Date	Topics	Things to turn in today
Mon 6/25	Exploring Data: csv files, csv python module, and importing csv files refresh CT concepts, practices and perspectives refresh Intro Visualizing Data plot (VD) project	1. Reading response 7 2. Daily note *Note: No ED showcase
Tue 6/26	Introduction to pandas and matplotlib Expressions and plots Python basics 1-2-3-4-5-6 practice Work on Visualizing Data (VD) project	1. Reading response 8 2. Daily note
Wed 6/27	Expressions and plots, cont. Python basics 1-2-3-4-5-6 practice Debug Challenge Work on Visualizing Data (VD) project	1. Daily note

Module 6: Data Storytelling

Contextualizing data and telling its story is as important as querying and plotting a computation. In this module, we'll extend our work from Modules 4 and 5, to narrate the story of the Marvel and DC Superhero Universes in a Jupyter Notebook.

Date	Topics	Things to turn in today
Thu 6/28	Exploring Jupyter Notebook markup language Refining data queries and plots File backups!	1. VD project 2. VD reflection 3. Daily note
Fri 6/29	Visualizing Data (VD) showcase Ethics of data Python basics 1-2-3-4-5-6 practice Intro Telling Stories (TS) project	1. Daily note
Mon 7/2	Python basics 1-2-3-4-5-6 practice Work on Telling Stories (TS) project	1. Reading response 9 2. Daily note
Tue 7/3	Python basics 1-2-3-4-5-6 practice Work on Telling Stories (TS) project	1. Daily Note
Wed 7/4	**NO CLASS - HAPPY FOURTH OF JULY!**	

Thu 7/5	Revisit creative expression and computational thinking ideas Work on Telling Stories (TS) project	1. Daily note
Fri 7/6	Telling Stories (TS) showcase Project debugging/refinements from showcase feedback before last submission	1. FINAL TS project 2. TS reflection 3. Final paper 4. Daily note

CU Boulder Policies

Accommodation for Disabilities

If you qualify for accommodations because of a disability, please submit to your professor a letter from Disability Services in a timely manner (for exam accommodations provide your letter at least one week prior to the exam) so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities. Contact Disability Services at 303-492-8671 or by email at dsinfo@colorado.edu. If you have a temporary medical condition or injury, see Temporary Injuries guidelines under the Quick Links at the Disability Services website and discuss your needs with your professor.

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, you must inform us of any conflicts with religious observances at least 2 weeks before an assignment is due or at least 2 weeks before you must miss class. We will adjust due dates of assignments so as not to interfere with your religious obligations.

See the campus policy regarding religious observances for full details.

Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, color, culture, religion, creed, politics, veteran's status, sexual orientation, gender, gender identity and gender expression, age, disability, and nationalities. Class rosters are provided to the instructor with the student's legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records. For more information, see the policies on classroom behavior and the student code.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to maintaining a positive learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct, discrimination, harassment or related retaliation against or by any employee or student. CU's Sexual Misconduct Policy prohibits sexual assault, sexual exploitation, sexual harassment, intimate partner abuse (dating or domestic violence), stalking or related retaliation. CU Boulder's Discrimination and Harassment Policy prohibits discrimination, harassment or related retaliation based on race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been subject to misconduct under either policy should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding sexual misconduct, discrimination, harassment or related retaliation can be found at the OIEC website.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the academic integrity policy of the institution. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access, clicker fraud, resubmission, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code Council as well as academic sanctions from the faculty member. Additional information regarding the academic integrity policy can be found at http://honorcode.colorado.edu.

Acknowledgements

The design of this course was influenced by many people and courses. This course builds on prior versions of INFO 1201 taught by CU Boulder Information Science Professors Ricarose Roque, Stephen Voida and Danielle Szafir. We use the Computational Thinking framework developed by MIT Professors Karen Brennan and Mitch Resnick (2012). We also took inspiration from ATLAS Professor Ben Shapiro's course ATLS 4519 Code Sorcery for New Wizards. Some of our course policies were developed from INFO 1111 Representations taught by Professors Jed Brubaker and Ricarose Roque.