

Course Syllabus

[Jump to Today.](#)

 [Edit](#)



CMN 152V – Social Science with Online Data

Instructor: Prof. Seth Frey

TA: Beril Bulat

(TA Zoom Office -by appointment-

<https://ucdavis.zoom.us/j/9644917511> [.\(https://ucdavis.zoom.us/j/9644917511\)](https://ucdavis.zoom.us/j/9644917511)

ID: [964 491 7511](https://ucdavis.zoom.us/j/9644917511) (tel:[9644917511](https://ucdavis.zoom.us/j/9644917511).)

Overview

Online social science methods use technology to push the bounds of what we can know about human interaction and communication. The digitization of communication has provided a wealth of data that is transforming the social sciences. But there's a gap between the important questions about human nature and the raw messy reality of code and data. In this class we will bridge the gap, learning enough of a popular programming language—Python—to discover things about ourselves, our friends, each other, and society, as we interact with the Internet in code. No prior skills are assumed.

All lectures and interactive assignments are online in this course. The course provides unit credit towards graduation on [all UC campuses](https://crossenrollcourses.universityofcalifornia.edu) [_\(https://crossenrollcourses.universityofcalifornia.edu\)_](https://crossenrollcourses.universityofcalifornia.edu), as well as UC Davis GE for 'Social Sciences', 'Quantitative Literacy', and 'Scientific Literacy' (for other UCs: please petition GE and major/minor credit with your advisor). This course is built around the 8 modules. Mid-term and final exams will be proctored online.

Learning objectives

By successfully completing this course, students should be

- Capable of thinking in code
- Able to write, skim, and tweak code
- Doing useful things on the Internet with code
- Inspired to want to explore and interact with society through code

Although you will do a lot of reading code, running code, and answering questions about code, you will do very little writing code from scratch. It is *not* an objective of this course to make you a programming expert. That is a job for other programming courses.

Class interaction

- For a group of people who you can struggle with and ask questions, you have a course group. You can message them through the *Canvas Inbox* (over to the ←).
- I will regularly use *Canvas Announcements* to send important updates. Be sure that you have them [activated](https://community.canvaslms.com/docs/DOC-10624-4212710344) [_\(https://community.canvaslms.com/docs/DOC-10624-4212710344\)_](https://community.canvaslms.com/docs/DOC-10624-4212710344) to appear in your email.
- Piazza will be the main venue for questions in the course. Piazza is a platform for managing discussions in Canvas. Just like it sometimes useful for you to listen in when others ask questions

in class, students find it useful to follow along on Piazza. You have to sign up the first time you use it. *You will have an opportunity to opt into Piazza Careers, which we do not use or promote due to its privacy policy.*

- To ask more personal questions about your status in the course, accessibility needs, or other specific concerns use Piazza to send a private message to the professor

speaking concerns, you are allowed to send a private message to the professor.

- TA Office hours or coordinate via Piazza.

Grading

The course will be graded out of 1000 points.

- **10 points (1% of total grade) SETUP**
 - Smaller tasks related to course administration or assignment prep
- **160 points (16% of total grade; 20 pts/wk) CODE LESSONS**
 - Eight code notebooks
- **110 points (11% total; 13.75 pts/wk) WEB SCIENCE DEMOS**
 - Demos will be provisionally spot checked for correctness of code edits.
 - Grading will assign
 - 10 points for implementing all requested small edits correctly in submitted webcam/screencap video
 - 2 points for showing face and student or official ID in the video
 - 2 points for submitting notebook file with the same edits as in the video
- **80 points (8% total; 10 pts/wk) CODE OR HYPOTHESIS GENERATION :**
 - Each assigned on alternating weeks
 - Hypothesis generation and reflection
 - Will be spot graded by hand by rubric
 - Code generation problem set
 - Will be graded for the correctness of code in the submitted notebook
 - These will be more difficult than other assignments. Fortunately, they are only 5% of your grade.
 - These assignments will alternate by week
- **40 points (4% total; 5 pts/wk) TOPIC LECTURES**
 - Lecture questions
- **600 points (60% of total grade) EXAMS**
 - **NOTE:** *The majority of your course grade will come from exam performance!*
 - Exams will be proctored
 - The exams are all multiple choice
 - Mid-term exam on Canvas, proctored 25%
 - Final exam on Canvas, proctored 35%
- **0 points (0%) ELECTIVE and EXTRA CREDIT**
 - Optional and ungraded work, and opportune feedback
 - There may be occasional opportunities for extra credit in this course

Canvas's Grades tab can be confusing. Your final grade will be the number of points you earned divided by 1000.

Course Content

Weekly Goals

Each week has four goals:

- Helping you put a programming concept into practice
- Illustrating the use of a web API to do something concrete on the Internet
- Exposing you to the habits of body and mind that make people prepared for programming
- Providing an insight into society or provoking curiosity about it

These goals track the learning objectives. Towards these goals, each week will include

- a code lesson
- a web science demo
- short topic lectures
- a task requiring either programming or scientific hypothesis generation.

You will start each week at the module page, and work down through the required items. Each item for submission will be marked on the Canvas Module page with a .

"Textbook"

Your main "textbook" will be in the form of "code notebooks" provided at no additional charge as part of the course. A code notebook is like a chapter of an interactive textbook. It mixes runnable, editable code with explanatory text that provides much more engagement than a textbook. I will distribute several notebooks for each module, some teaching you an elementary code concept "from the bottom up", others walking through a code application "from the top down".

Course groups

To make the course more personal, you will be grouped with other students into small groups. Your discussion activity will be restricted to these students, and you can use these peer groups as a good first resource for questions on assignments.

Course Activities

Code ecosystem topic lectures

- **Format.** The code ecosystem topic lectures will be short video lectures. You will watch a few 1-3 min videos per week and answer questions about them.
- **Purpose.** These lectures will complement the "hard skill" programming pedagogy in other course content with "soft skill" insights into programming practice.
- **Assessment.** This content will be assessed with tonic questions embedded into video content

Assessment: The content will be assessed with topic questions embedded into video content. Questions about these lectures will also appear on the mid-term and final exams.

Code lesson

- **Format.** Each lesson will be presented in an interactive code notebook and accompanied by a video walkthrough. You will watch the video and work through each notebook by answering questions embedded into it.
- **Purpose.** The code lesson will teach basic programming concepts.
- **Assessment.** Embedded in the code notebooks will low-stakes (multiple choice, multiple attempts) interactive micro-assessments, for fast specific feedback. Complete notebooks will be submitted and graded for completion and correctness. Questions drawn from the micro-assessments will constitute 80% of the mid-term and final exams, so make sure that you understand your answers to these questions. It's not enough to have all the right answers in your lessons, you have to know why it's the right answer. *Tip: as you work through each lesson, keep track of which questions you got right for unknown reasons, so you can ask about them on Piazza.*

Web science demos

- **Format.** Each demo will be presented in a code notebook and accompanied with a video walkthrough. You will watch the video, run the demo, and record a video of yourself making and running specific tweaks to the notebook.
- **Purpose.** The web science demo will show complex code in action performing useful tasks in interaction with the Internet. These will also provide data-driven insights into a social or cultural phenomenon.
- **Assessment.** Each will be assessed with brief webcam/screencast video submissions showing each student's ability to edit key demo functionality on the fly. Questions about these demos will also appear on the mid-term and final exams.

Code Generation and Hypothesis Generation

- **Format.** Every other week, students will either generate code in problem sets, or hypotheses about web data. The problem sets will be administered in code notebooks. The hypothesis generation exercises, administered in Canvas, will elicit student's reasoning about social data from the week's content, ask students to generate alternative hypotheses to explain some aspect of the data, and suggest supplementary data that would distinguish those hypotheses.
- **Purpose.** The hypothesis generation exercises will give students experience thinking about online data and its relationship to scientific inquiry. The coding exercises will test your ability to author code from scratch. They are a small part of the course, but will give you your first experiences actually writing code, rather than just reading it, running it, or answering multiple choice questions about it

- **Assessment.** The programming exercises will be submitted and automatically graded for correctness, completion, and correct manner of completion. The hypothesis generation exercises will be manually graded by rubric

Exams

[See here](#) for full information about the exams.

- **Format.** There will be a mid-term and a final exam, both consisting of multiple-choice questions, both an hour long. The content of these exams will be drawn primarily (80%) from the example questions embedded into each code lesson. There will also be some additional questions about the demo, topic, and problem set content. The first will cover the first four weeks of content. The second will cover the remaining weeks, but some tests of cumulative knowledge (over all modules) will be inevitable.
- **Purpose.** The mid-term and final will provide summative assessment of each of the above, with a focus on the lesson content's basic programming pedagogy.
- **Assessment.** Both exams will be proctored using online proctoring service Examity.

General Policies

Due dates and late submissions

Assignments are due Sundays at 8:55PM PST. Late submissions are still welcome. You will be penalized 15 % each late day until you reach 55%. After that, you can maximally receive 55% for all submissions until the Sunday of the last course Module. This means that if you did perfect on all course content, but put it all off until the last week, it would not be possible to pass the course.

There is one big downside to submitting late work. Students who submit on time are guaranteed to get fast feedback and grades. Students who submit late may not receive grades or feedback until the end of the term. Since quality feedback is an important part of learning and exam preparation, students who chronically submit late work may be at a disadvantage during exams.

Spot checking

Some assignments are graded on spot checks. Under spot checking, most submissions by most students will provisionally get full credit, but some will be selected randomly for close grading against a rubric. The upside of provisional full credit is that you get full credit by default. The downside is that it is provisional: the instructors reserve the right to reconsider the default grade at any point in the term and grade your submission against the rubric. We can also review your previous submissions and re-evaluate assignments from past modules that received the default maximum grade. Under this policy, you should always submit on the assumption that you will be graded against the rubric, either in that week or future weeks. Low-quality submissions may slip through the cracks for a few weeks, but the risk that they will be re-evaluated returns every week, making it unlikely to succeed. We use this policy so that we can continue to provide quality feedback on your graded submissions despite

this policy so that we can continue to provide quality feedback on your graded submissions despite the size of the course.

Cheating and collaborating

General. You are responsible for abiding by the [UC Davis Code of Conduct](https://ossja.ucdavis.edu/code-academic-conduct) (<https://ossja.ucdavis.edu/code-academic-conduct>), as well as the code for your own university, if you are enrolled from another campus. Any suspicion of cheating of any kind will be reported to Student Judicial Affairs. **Cheating will result in an 'F' in the course.** Cheating includes (but is not limited to): turning in work that is not yours, sharing answers or posting them online, sharing information during exams, taking photos of exams, discussing exams with late takers, having possession of or copy of a stolen exam (includes exams posted to work-sharing sites), taking exams for others, etc. You are expected to turn in your own work. Plagiarism will not be tolerated. If you use the work of another, you must appropriately cite that work (for example, I drew this policy from Dr. Jeanette Ruiz). In general, if you feel a temptation to cheat, you should ask why you are in college, and consider taking time off.

Code-specific guidance: Collaborating. In the world of code, cheating can be less black-and-white. I encourage you to discuss course activities with your friends and classmates as you are working on them, to ask for help on specific questions, and to study together for exams, whether in person or virtually. It is a great idea to join forces (in personal meetings or through online collaboration) to identify gaps in your understanding. You will definitely learn more in this class if you work with others than if you do not. Ask questions, answer questions, and share ideas liberally.

If you are helping another student, don't just tell them the answer; they will learn very little and run into trouble on exams, which are most of the grade. It is not enough to find the right answer, you have to understand it. Instead, try to guide them toward discovering the solution on their own. The problem-solving practice is the key to progress in code. Any collaboration is highly counterproductive when you start depending on it.

As for notes and third-party study guides, experience shows that only those students succeed who actively do the assignments and make their own notes. This is because it is not the study notes, but the interactive process of preparing notes that makes the difference!

Code-specific guidance: Cheating. Of course, cooperation has a limit, and that limit is sharing code or answers.

- Do not engage in behaviors that violate the [Code of Conduct](https://ossja.ucdavis.edu/code-academic-conduct) (<https://ossja.ucdavis.edu/code-academic-conduct>) for this course.
- Do not copy code or answers from others.
- Do not let others copy code or answers from you, or otherwise turn in work that you did.
- Do not copy full solutions from online sources such as Stack Overflow, Pastebin, and GitHub. Smaller snippets of solutions may be acceptable.
- Do not post your solutions publicly during or after the term.
- Do not search for publicly posted solutions to this course, during or before the term.
- Don't show your code to other students, except to someone who has already submitted the assignment and is helping you finish.

Again, cheating will result in an 'F' in the course.

Religious observances

If you have a religious observance that conflicts with your participation in the course, please meet with me before the end of the second week of the term to discuss appropriate accommodations.

Student accessibility needs

Access is a right that all students have. The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities.

Students requiring disability-related academic adjustments and services may consult the UC Davis Student Disability Center. For those with documented testing accommodations, please have your accommodations letter emailed to me ASAP. Also, it is your responsibility to contact me, at least one week before each exam to arrange to take your tests at an alternate location/for longer test periods. Tests must be taken on the same day as the rest of the class.

I recognize that students may have health, mental, or physical issues and concerns beyond those recognized by university services. I encourage you to ask for what you need in order to be successful in the course.

FAQs (important: please read!)

- ***Will the exams cover mostly lecture material or labs?*** Answer: Exams draw 80% from the example questions in the code lessons. The remaining 20% will be from the other graded course content (such as topic lectures and web science code demos).
- ***Will there be tasks and questions that require any extra programming skills and background in math or statistics?*** Answer: No, no prerequisites required. If you pay attention and do the readings, there is absolutely no reason why not to achieve an A+. You got a high school degree and made it here, we trust you can do it.
- ***Are the intermittent questions in the video lectures graded?*** Answer: Yes. They are worth half of a letter grade (5%). Exams will include questions that you are unlikely to answer correctly if you haven't actively watched the lectures and taken the intermittent questions seriously. Not doing them right, will hurt your grade twice. If you got one wrong, go back immediately and re-watch that segment to be prepared for the exam. The questions on the exam are NOT literally the same questions, so please don't waste time memorizing them. Understand what they are about.
- ***How do I know if I got an intermittent video question wrong/right in PlayPosit?*** Answer: We use a platform called PlayPosit to embed questions in lecture content. If you marked a question correctly it turns green (or stays white for correctly not marking it). Wrong answers turn yellow/reddish.
- ***The PlayPosit interface shows that I got points on the questions. but it doesn't show up in***

- Canvas?** Something went wrong with the sync process (maybe an internet connection problem...). The player always tries to sync grades AFTER reaching the end of the video. If that effort fails, then the Sync Grades button should show up. You can refresh the page and open up the Review tab again to click on the newly re-enabled Sync Grades button. Please ALWAYS CHECK at the end of your Session if all grades got synced. It is always your responsibility (here and elsewhere) to check if the tech you are using also worked.
- **Will the final exam be accumulative (cover material from the first part of the course)?**
Answer: No, it covers modules 5–8. But: some concepts will follow us around during the entire course.
 - **I have anxiety during exams, especially during multiple-choice tests. What can I do?**
Answer: This is a serious disadvantage, but one you have to learn to confront.
 - I encourage you to visit the UC Davis [Student Academic Success Center](http://success.ucdavis.edu) (<http://success.ucdavis.edu>) or the equivalent at other UC campuses.
 - I also encourage you to get a better understanding of effective learning techniques. For example, see this popular online course on [Learning How to Learn](https://www.coursera.org/learn/learning-how-to-learn) (<https://www.coursera.org/learn/learning-how-to-learn>) at UCSD.
 - Students that require accommodations can request more time to complete the exams.
 - **The videos are lagging, what can I do?**
Answer: By default, streaming is set to a full HD version and depending on the network connection/bandwidth, you may experience the lagging/pausing. You can adjust video quality from the scrub bar (<HD>) or adjust the video course ([more here](https://knowledge.playposit.com/article/212-the-student-experience-3-0) (<https://knowledge.playposit.com/article/212-the-student-experience-3-0>)) as that should help. You can also watch the videos (without questions) here on the [course YouTube channel](https://www.youtube.com/channel/UCILw2teg04tJX3tQrzlaFQA) (<https://www.youtube.com/channel/UCILw2teg04tJX3tQrzlaFQA>).
 - **We are required to upload videos for some assignments, but mine won't upload?**
Answer: Shortly before the assignment deadline, we collectively overload the server sometimes. This is a real risk for any digital work, please prepare for it. If you do not want to run the risk of getting the first-day 15% late-discount, please make sure to finish your assignment before Sunday afternoon. This also gives you wiggle-room in case of any other technological problem, and you have to expect technological problems when doing computational social science.
 - **What communications / messages should I get in order to be on top of what's going on?**
Answer: Through Canvas (our main platform) you will get one Welcome email from the instructor per Module and sometimes general feedback for assignments from TAs. These are IMPORTANT! Make sure you have your Canvas notifications settings **TURNED ON** (<https://community.canvaslms.com/docs/DOC-10624>)!

Your Instructor

Seth Frey is an Assistant Professor at the University of California, Davis in the Department of Communication, a part of the Computational Communication group. On the way to this position, he had time at Disney Research, a part of Walt Disney Imagineering, where he applied my expertise to both theoretical and practical questions about engineered social systems. He was a fellow at Dartmouth College's Neukom Institute for Computational Science. He is from Sunnyvale, CA and <https://canvas.ucdavis.edu/courses/511847/assignments/syllabus>

Bartholomew College & Research Institute for Computational Science. He is from Sunnyvale, CA and graduated from Cal in 2004.

Course Summary:

Date	Details
Sun Sep 27, 2020	 SSOD 1 Code Demo "LANGUAGES" https://canvas.ucdavis.edu/courses/511847/assignments/535911 due by 8:55pm
	 SSOD 1 Code Lesson "CODE NOTEBOOKS" https://canvas.ucdavis.edu/courses/511847/assignments/535912 due by 8:55pm
	 SSOD 1 Hypothesis Generation Exercise "TRANSLATION QUALITY" https://canvas.ucdavis.edu/courses/511847/assignments/535905 due by 8:55pm
	 SSOD 1 SETUP Group Introductions https://canvas.ucdavis.edu/courses/511847/assignments/535906 due by 8:55pm
	 SSOD 1 SETUP Proctor Registration https://canvas.ucdavis.edu/courses/511847/assignments/535913 due by 8:55pm
	 SSOD 1 SETUP Screen Capture https://canvas.ucdavis.edu/courses/511847/assignments/535914 due by 8:55pm
	 SSOD 1 SETUP Self-Surveillance With Privacy Badger https://canvas.ucdavis.edu/courses/511847/assignments/535915 due by 8:55pm
	 SSOD 1 SETUP Syllabus Quiz https://canvas.ucdavis.edu/courses/511847/assignments/535904 due by 8:55pm
	 SSOD 1 Topic 1 "The point of this course" https://canvas.ucdavis.edu/courses/511847/assignments/535916 due by 8:55pm
	 SSOD 1 Topic 2 "Why code?" https://canvas.ucdavis.edu/courses/511847/assignments/535917 due by 8:55pm

Date	Details
Sun Oct 4, 2020	 SSOD 1 Topic 3 "Generating hypotheses and conceiving tests" due by 8:55pm
	 SSOD 2 Code Demo "KITTENS" due by 8:55pm
	 SSOD 2 Code Generation Problem Set "PIZZA IN METRIC" due by 8:55pm
	 SSOD 2 Code Lesson "PYTHON AS CALCULATOR" due by 8:55pm
	 SSOD 2 SETUP Twitter "Keys" due by 8:55pm
	 SSOD 2 Topic 1 "Thinking in code" due by 8:55pm
	 SSOD 2 Topic 2 "Succeeding in an online course" due by 8:55pm
	 SSOD 3 Code Demo "TWITTER API" due by 8:55pm
	 SSOD 3 Code Lesson "PYTHON HAS DIFFERENT KINDS OF THINGS" due by 8:55pm
	 SSOD 3 Hypothesis Generation Exercise "TWEET GEOLOCATION" due by 8:55pm
Sun Oct 11, 2020	 SSOD 3 Topic 1 "Getting coding help" due by 8:55pm
	 SSOD 3 Topic 2 "Feeling worthy to code" due by 8:55pm

Date	Details
Sun Oct 18, 2020	 <u>SSOD 4 Code Demo "WEB HISTORY"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535929 due by 9:55pm
	 <u>SSOD 4 Code Generation Problem Set "TEETHING"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535930 due by 9:55pm
	 <u>SSOD 4 Code Lesson "COLLECTIONS OF THINGS IN PYTHON"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535931 due by 9:55pm
	 <u>SSOD 4 Topic 1 "How to prepare for the exams"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535932 due by 9:55pm
Thu Oct 29, 2020	 <u>Midterm Exam Proctored</u> https://canvas.ucdavis.edu/courses/511847/assignments/535899 due by 9pm
	 <u>Midterm Exam_v1</u> https://canvas.ucdavis.edu/courses/511847/assignments/535903 due by 11:59pm
	 <u>Midterm Exam_v2</u> https://canvas.ucdavis.edu/courses/511847/assignments/535902 due by 11:59pm
Sun Nov 1, 2020	 <u>SSOD 5 Code Demo "WIKIPEDIA"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535933 due by 8:55pm
	 <u>SSOD 5 Code Lesson "CODE FLOW"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535934 due by 8:55pm
	 <u>SSOD 5 Hypothesis Generation Exercise "FAME"</u> https://canvas.ucdavis.edu/courses/511847/assignments/535908 due by 8:55pm
	 <u>SSOD 5 SETUP Instagram Keys</u> https://canvas.ucdavis.edu/courses/511847/assignments/535935 due by 8:55pm
	 <u>SSOD 5 SETUP Twilio Keys</u> https://canvas.ucdavis.edu/courses/511847/assignments/535936 due by 8:55pm

Date	Details	
	 SSOD 5 Topic 1 "Thinking through code flow" https://canvas.ucdavis.edu/courses/511847/assignments/535937	due by 8:55pm
	 SSOD 5 Topic 2 "How to debug code" https://canvas.ucdavis.edu/courses/511847/assignments/535938	due by 8:55pm
Fri Nov 6, 2020	 Your feedback on the course https://canvas.ucdavis.edu/courses/511847/assignments/535952	due by 8:55pm
	 SSOD 6 Code Demo "GUTENBERG" https://canvas.ucdavis.edu/courses/511847/assignments/535939	due by 8:55pm
	 SSOD 6 Code Generation Problem Set "FILTERING LISTS" https://canvas.ucdavis.edu/courses/511847/assignments/535940	due by 8:55pm
Sun Nov 8, 2020	 SSOD 6 Code Lesson "DOING ONE THING IF ANOTHER" https://canvas.ucdavis.edu/courses/511847/assignments/535941	due by 8:55pm
	 SSOD 6 Topic 1 "How to skim code" https://canvas.ucdavis.edu/courses/511847/assignments/535942	due by 8:55pm
	 SSOD 7 Code Demo "INSTAGRAM" https://canvas.ucdavis.edu/courses/511847/assignments/535943	due by 8:55pm
	 SSOD 7 Code Lesson "DICTIONARIES AND NESTED DICTIONARIES" https://canvas.ucdavis.edu/courses/511847/assignments/535944	due by 8:55pm
Sun Nov 15, 2020	 SSOD 7 Hypothesis Generation Exercise "ALGORITHMIC BIAS" https://canvas.ucdavis.edu/courses/511847/assignments/535909	due by 8:55pm
	 SSOD 7 Topic 1 "Copying from code on the Internet" https://canvas.ucdavis.edu/courses/511847/assignments/535945	due by 8:55pm
	 SSOD 7 Topic 2 "Hiding from the complexity" https://canvas.ucdavis.edu/courses/511847/assignments/535946	due by 8:55pm

Date	Details
Sun Nov 22, 2020	 SSOD 8 Code Demo "ROBOCALLS" https://canvas.ucdavis.edu/courses/511847/assignments/535947 due by 8:55pm
	 SSOD 8 Code Generation Problem Set "TURTLE GRAPHICS" https://canvas.ucdavis.edu/courses/511847/assignments/535948 due by 8:55pm
	 SSOD 8 Code Lesson "ADDING FUNCTIONALITY TO CODE" https://canvas.ucdavis.edu/courses/511847/assignments/535949 due by 8:55pm
	 SSOD 8 Topic 1 "Reading documentation" https://canvas.ucdavis.edu/courses/511847/assignments/535950 due by 8:55pm
	 SSOD 8 Topic 2 "Becoming a good coder" https://canvas.ucdavis.edu/courses/511847/assignments/535951 due by 8:55pm
Thu Dec 10, 2020	 Final Exam_Proctored https://canvas.ucdavis.edu/courses/511847/assignments/535900 due by 9pm
	 Final Exam_v1 https://canvas.ucdavis.edu/courses/511847/assignments/535898 due by 9pm
	 Final Exam_v2 https://canvas.ucdavis.edu/courses/511847/assignments/535901 due by 9pm
	 Additional Adjustment https://canvas.ucdavis.edu/courses/511847/assignments/535910