Course Information for ECS 170 – Introduction to Artificial Intelligence
Spring Quarter 2022 (v. 2 - updated March 31)

Instructor:

Kurt Eiselt
eiselt@ucdavis.edu
Kemper 3051

Teaching Assistants:

Fangzhou Li fzli@ucdavis.edu
Dom Huh dhuh@ucdavis.edu

Lectures:

The pandemic may not be over yet, but we expect to be in the classroom for the entire quarter. There is no online option for this course. Unless the university administration tells us otherwise, our lectures will be in person, following this schedule:

TR 9:00 – 10:20 AM GIEDT 1003

Note that the university may require us to pivot back to online teaching with little notice.

Discussion Section:

Note: The discussion section will be held in person.

T 11:00 – 11:50 AM SCRBOK 160

Office Hours:

Note: no office hours during first week of classes. Office hours for week 2 and beyond may be held via Zoom.

Kurt Soon to be determined
Fangzhou Soon to be determined
Dom Soon to be determined

Catalog Description:

Design and implementation of intelligent computer systems. Knowledge representation and organization. Memory and inference. Problem solving. Natural language processing.

Objectives:

Students will:

- learn some of the fundamental principles of artificial intelligence
- implement some of these principles in Python

Prerequisites:

Official: ECS 60 or ECS 32B or ECS 36C

Unofficial:

- ability to read and write moderately-sized Python programs
- familiar with data structure concepts and applications, especially breadth-first search, depth-first search, trees, and graphs
- competence with recursion

Textbook:

Artificial Intelligence: A Modern Approach (Fourth Edition) by Stuart Russell and Peter Norvig. Pearson Education, 2020.

Very Very Very Tentative and Approximate Schedule (likely to change)

Week	Topics	Chapter reading
1	Introduction	Chapters 1 & 2
2 & 3	Problem Solving and Search	Chapters 3, 4, & 5
4 & 5	Knowledge, Reasoning, and Planning	Chapters 7, 8, 9, 10, & 11
6 & 7	Learning	Chapters 19 (maybe 21)
8	Natural Language Processing	Chapter 23
9 & 10	Philosophy and Ethics	Chapter 27

Grading (we might modify this slightly):

- *Five homework (programming) assignments (30% of final grade)*
- *Four in-person quizzes (20% of final grade) – **note change to in-person***
- *One midterm exam (20% of final grade)*
- *One final exam (30% of final grade)*

Exam Dates (midterm date subject to change based on class progress):

Midterm Tuesday, May 3

Final Exam Friday, June 3, 1:00 – 3:00pm

Exams for students with accommodations will take place at the same days and times listed above, but in a different location to be announced.

Late Homework and Makeup Exams:

We do not accept late homework except in the case of documented illness, documented family emergency, or documented university-approved absence. Forgetfulness is not an accepted excuse for late homework. Submitting your homework to the wrong repository on Canvas, even if submitted before the deadline, is not an accepted excuse for late homework.

There will be no makeup midterm exam. If you miss the midterm exam due to documented illness, documented family emergency, or documented university-approved absence, your midterm exam score will be computed from the score you receive on the final exam. There is no makeup final exam available during spring quarter. If you miss the final

exam due to documented illness, documented family emergency, or documented university-approved absence, you will receive a grade of incomplete and the final exam will have to be taken at some later date.

If you miss any exam for some reason other than documented illness, family emergency, or university-approved absence, your exam score will be zero.

Submitting Homework for Evaluation:

Your solutions to homework assignments must be submitted through Canvas by the indicated day and time. We will give a few minutes of grace period to accommodate differences between your clock and ours, but don't expect more than a few minutes. If you have not submitted your work before the grace period has expired, you will not get credit for the work.

Submit your solution early, in case of problems. Don't wait until the last minute. Make sure you submit the correct file. We won't accept late homework, even if you had difficulty with Canvas or you turned in the wrong thing.

Any programs must be written using Python 3. Programs written in earlier versions of Python, or in any language other than Python 3, will not receive credit.

We do not accept e-mailed homework.

Regrades:

Requests for regrades because of perceived grading errors must be submitted within one week of homework evaluations being returned. No requests for regrades will be accepted after that week has passed.

UC Davis's Code of Academic Conduct

You are expected to have read UC Davis's Code of Academic Conduct, which you may find here: https://ossja.ucdavis.edu/code-academic-conduct

If you have questions about the Code of Academic Conduct, please ask your instructor or contact the Office of Student Support and Judicial Affairs.

Additional Thoughts about Academic Misconduct:

As the instructor and teaching assistants, it is our responsibility to make tests and assignments that are fair, to grade fairly, to look for cheating, and to refer students who cheat to Student Judicial Affairs for possible sanctions. Students in ECS classes are most often referred to OSSJA for copying each other's programs or copying on tests.

As students, it is your responsibility to avoid cheating and to discourage other students from cheating.

It is pretty clear what it means to cheat on a test, in this class just like any other. It is sometimes less clear to students when they are cheating on a programming assignment. We want you to help each other, and we want you to look at examples of similar programs. So how do you know when helping and looking crosses the line into cheating? Here's our basic rule:

You should write each line of your program yourself, and you should know what it does and why it is there.

To check yourself, you should type each line of your program yourself; *never copy or cut-and-paste from another file*. While you're typing, ask yourself, do I know what this line does? Why am I including it in this program?

Often the easiest way to write a program, even in industry, is to take an already-existing program that does something similar, and change it around. This is fine in "real-life", but in this class it is better not to cut-and-paste whole programs or even single lines, since as a beginner you need to concentrate on every line. You should, however, look very carefully at the example programs from lecture or the textbook, and figure out how your programs should be similar or different.

Writing programs can be very, very frustrating. Sometimes you don't know how to start. Sometimes your program seems perfect, but it doesn't do what you think it should be doing. We want you to talk to the other students, to friends who are programmers, to the TA, to the instructor, to anyone who can help! We want you to show them your programs and ask them what's wrong. But make sure when the conversation is over that you understand every line of your (hopefully improved) program. If your friend is telling you exactly what to type, you are cheating.

If you are looking at another student's program to help him or her, that is not cheating. If you are showing another student your program to help him or her, that is cheating. If you are looking at another student's program while typing in yours, or if you cut-and-paste from another student's program, that is obviously cheating and you are very likely to be caught. Two (or more) students who work together on a single solution and then submit separate copies of that solution have crossed the line into cheating. A student who copies a solution that was provided by a tutor or that was found on the Internet has crossed the line into cheating.

You will quickly see that there are always many, many ways to write a program for a particular programming assignment, just like there are many ways to write an essay for a particular English assignment. And it is almost as easy to recognize two almost identical programs as it is to recognize two almost identical essays. For some of the assignments in this

course, we will use software to detect almost identical programs. If you find yourself changing around someone else's program to try to fool the detection software, you are cheating.

One other thing: a given homework assignment is worth only a small fraction of your final grade. UC Davis's Code of Academic Conduct gives faculty the authority to assign a final course grade of "F" in cases of academic misconduct. So, when you cheat on the homework, you're risking a failing grade in the course (and maybe worse) for that relatively small fraction of your grade. You need to ask yourself if that's a risk worth taking. Furthermore, if you can't do the homework on your own, it's highly unlikely that you'll get passing grades on the exams.