General Information

Instructor

Matt Bishop	*Email*: mabishop@ucdavis.edu
Office: 2209 Watershed Science	*Web*: http://seclab.cs.ucdavis.edu/∼bishop
Office Hours: MWF 11:00am–11:50am	*Phone*: (530) 752-8060

When you send me email, please begin the Subject field with "ECS 36A" so I see that the letter has to do with the class. I receive lots of email and, while I look at it all, I *sometimes* miss things, or skim the Subject fields to see which letters are very important. Putting "ECS 36A" in the Subject field will tell me it is important.

Teaching Assistants

Iman Chatterjee, *Email*: imnchatterjee@ucdavis.edu
Yongshuai Liu, *Email*: yshliu@ucdavis.edu
Nikhil Wadhwa, *Email*: nwadhwa@ucdavis.edu

TA Office Hours

Monday 4:10pm–6:00pm in 47 Kemper (Yongshuai Liu)
Tuesday 10:00am–12:00pm in 55 Kemper (Nikhil Wadhwa)
Wednesday 11:50am–1:00pm in 47 Kemper (Yongshuai Liu and Iman Chatterjee will alternate for these)
Wednesday 12:00pm–2:00pm in 55 Kemper (Iman Chatterjee and Yongshuai Liu will alternate for these)
Thursday 10:30am–12:00pm in 47 Kemper (Nikhil Wadhwa)
Friday 3:10pm–5:10pm in 53 Kemper (Iman Chatterjee)

Lectures

MWF 10:00am–10:50am in 100 Hunt

Discussion Sections

You may go to any discussion section you like as long as there is room for those registered for that section.

Section A01	M 1:10pm–2:00pm in 1204 Haring
Section A02	W 1:10pm–2:00pm in 204 Art
Section A03	F 2:10pm–3:00pm in 146 Olson

The TAs will rotate doing discussion sections, each taking all three sections in a week.

Course Outline

Computers and computer programming for students with some prior experience, algorithm design, and debugging. Good programming style. Use of basic UNIX tools.

Course Goals

The overall goal is to learn computers by studying programming and how to use them to solve problems. More specifically, we hope you will:

1. Learn how to use the Linux (and UNIX) systems;
2. Learn how to write programs, and how to use an integrated development environment, debuggers, compilers, and interpreters;
3. Learn the basics of the Python programming language, and through it the basic control and data structures, operations and data types in programming languages; and
4. Learn how to design and write an algorithm.

Prerequisite

Prior experience with basic programming concepts (variable, loops, conditional statements) required; you must satisfy the computer science placement exam, or C- or better in ECS 32A.

Text

Required: Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers, *How to Think Like a Computer Scientist: Learning with Python 3 Documentation*, 3rd Edition (2018). It is on-line at `https://media.readthedocs.org/pdf/howtothink/latest/howtothink.pdf` and is free.

Required: William E. Schotts, Jr., *Linux Command Line: A Complete Introduction*, No Starch Press, San Francisco, CA, USA (2012). It is on-line at `http://www.solutionsproj.net/software/The_Linux_Command_Linex.pdf` and is free.
ISBN: 978-1-59327-389-7.

Recommended: Allen Downey, *Think Python: How to Think Like a Computer Scientist*, 2nd Edition, Version 2.2.23, O'Reilly Media Inc., Sebastopol, CA, USA (2015). It is on-line at `http://greenteapress.com/thinkpython2/thinkpython2.pdf` and is free.
ISBN: 978-1-4919-3936-9.

Class Web Site

The class web site is on SmartSite. To access it, go to `http://smartsite.ucdavis.edu` and log in using your campus login and password. Then go to ECS 36A in your schedule. Announcements, assignments, handouts, and grades will be posted there, and you *must* submit any assignments there. The alternate web site, `http://nob.cs.ucdavis.edu/classes/ecs36a-2019-01` has everything except grades, and you cannot submit work there.

Extra Credit

Extra credit is tallied separately from regular scores. If you end up on a borderline between two grades at the end of the course, extra credit will count in your favor. But failure to do extra credit will never be counted against you, because grades are assigned on the basis of regular scores. You should do extra credit if you find it interesting and think that it might teach you something. Remember, though, it is not wise to skimp on the regular assignment in order to do extra credit!

Grading

Homework assignments	40%
Midterm	30%
Final	30%

Important. The grade E-NWS (sometimes called NWS or NS), which stands for "No Work Submitted", is *no longer a valid grade*. In cases where it would have been assigned in the past, we will give a grade of "F". Please be sure you *drop* this class rather than submit no work! This will also release a space so someone on the wait list can get in, so by dropping the class, in addition to avoiding a grade of F, you will help a student who is undoubtedly *very* concerned she or he will not get into this class.

Important Dates

First day of instruction: January 7, 2019
10-day drop deadline: January 18, 2019
Last day to add: January 23, 2019
Midterm exam: February 6, 2019
Last day to opt for P/NP grading: February 11, 2019
Last day of instruction: March 15, 2019
Final exam: March 21, 2019 from 1:00pm to 3:00pm

PTA Numbers

The department policy on issuing PTAs is available at `http://www.cs.ucdavis.edu/blog/pta-policy/`. If you need a PTA, please read that page, and follow the instructions there.

Academic Integrity

The UC Davis Code of Academic Conduct, available at `http://sja.ucdavis.edu/files/cac.pdf`, applies to this class. For this course, all submitted work must be your own. You may discuss your assignments with classmates or the instructor to get ideas or a critique of your ideas, but the ideas and words you submit must be your own. Unless *explicitly* stated otherwise, collaboration is considered cheating. Also, remember to cite, and give the source for, anything you copy or paraphrase, as is standard academic protocol. Plagiarism, even (especially) copying code from a book or the web without crediting it, is cheating.

The single exception to the rule against collaboration is debugging. *Once you have written your program*, if you need help debugging it, you are free to ask a classmate for help *providing that classmate has also written the program*. (This should avoid any unintentional copying.) Sometimes having someone else look over a program that is not quite working right will lead you to the best way to fix it, and you both will gain valuable experience in looking at programs and figuring out what is going on. But you must not collaborate on writing the program.

Any cheating will be reported to the Office of Student Support and Judicial Affairs.