EE 021: Introduction to Electrical Engineering Programming

Instructor: Ayush Pandey (meet with me 1-on-1 using this URL)

Email: ayushpandey@ucmerced.edu Office: SE2 381

Instructional staff:

- **Teaching assistant:** Shariful Islam. Email: sharifulislam@ucmerced.edu
- **Course reader:** Chandra Govindarajan (cgovindarajan@ucmerced.edu).
- **Learning assistants:** Max Fu (mfu27@ucmerced.edu), Jayesh Chaudhary (jchaudhary@ucmerced.edu), Kavin Rajasekaran (krajasekaran@ucmerced.edu)

🏛️ **Class Discussions:** Mon and Wed from 2pm to 3.15pm at COB1 120

📓 **Ayush's office hours:**

1. Mon from 3.30pm to 4.30pm in SE2 381 and SE2 390
2. Wed from 3.30pm to 4.30pm in SE2 381
3. Wed from 5.30pm to 6.30pm in SE2 381

🙋🏾 **Labs:**

02L: Thursday 5.30pm to 7.20pm in Admin 350
03L: Thursday 11.30am to 1.20pm in Admin 364
04L: Tuesday 2.30pm to 4.20pm in COB1 276

🙋🏾 **Group debugging sessions:**

1. Jayesh Chaudhary: Tue from 1pm to 2pm at SE2 Lobby
2. Kavin Rajasekaran: Mon 4 - 5pm in SE2 390
3. Max Fu: Fri from 2pm to 3pm at SE2 Lobby

TL;DR (too long, didn't read) summary of the syllabus:

- You will learn computer programming (primarily in Python) in this course, and you will be able to develop functioning and meaningful programs for engineering systems!
- **Bring your laptop and electronic supplies** to every class to participate in learning together with everyone.
- Contact Shariful (your TA) for any extension requests and make the most of LA office hours! We are here to support your learning and ensure that you can excel in this course.

Course prerequisites:

All UC Merced students are welcome! There are **no pre-requisites** to take this class and excel in it! It is expected that all students will be comfortable with middle school algebra and in using computers. This is intentionally defined in a subjective manner to welcome students from all backgrounds and experiences. This course serves as a training platform for all students interested in learning about computing and computer programming. This course will actively attempt to

create a welcoming climate for everyone without relying on prior computer programming experience. However, if you do have prior experience in any of the course topics, you will find many opportunities to grow and create advanced computing systems. If needed, feel free to discuss your preparation with the instructor or the course staff.

Course learning goals and outcomes:

By the end of this course, you will be able to demonstrate computer programming skills in at least two programming languages (including Python). You will be able to create custom programs, investigate errors in existing programs independently, and create fully functioning applications.

Course overview:

This course is your first introduction to "computing" in engineering. You will learn “Python" — a computer programming language to control what computers can do. A special twist in EE 021 is that you will learn to control not just the outputs from your computer but also control hardware so that you learn the applications of programming in the real world. The topics in this course are some of the most fundamental things that you will ever learn! Computing and programming literacy is increasingly becoming one of the most desirable skills, no matter what kind of education or job you are involved with in the future.

The philosophy of the course is grounded in the computing for all mission (CS for All, NSF CS for all report). The course staff will strive to foster learning opportunities such that every student can achieve an A in this course! The assessments will be designed in a way that this is possible and there will be no grading on a curve. The course staff will try their best to support your learning throughout the semester to lower the barriers in learning computer programming.

Catalog description:

Introduces students to programming and the logical thinking required to use electrical inputs to control a process. Concepts related to data types (e.g. variables, strings, arrays), logic statements, functions, and graph creation. Emphasizes Python but will also introduce other programming tools. A key focus is for students to be able to write programs to control circuits.

Course policy and expectations:

1. **Lectures** will be designed as class discussions where you will be able to learn most of the key concepts required to complete the assignments successfully. Class activities will be designed to foster an enjoyable learning experience (rather than just passively listening to the instructor :)
2. You **must bring: a laptop and electronic supplies (will be loaned to you by the school).** Your laptop device will be useful to write and run code (refer to the school suggested specifications) and electronic components to create circuits.
3. The electronic components will be distributed during classes, as needed. You must bring all previously distributed components to each class to successfully participate in the class.

At the end of the semester, you are required to return all electronic components to the school.

4. **Lab attendance is mandatory.** A lab checkout assignment will be due every week, which you must get checked off by the TA in-person in the lab. The lab check-out assignment will be a small part of the main weekly assignment. The requirement to go to the lab is enforced so that you get to meet other students in the class and form a community of your peers to study together and also to connect you with the help that you may need to succeed in the course. At the same time, the checkout assignment is your way to take an early leave from the lab, if you prefer.
5. Try to make the best of the **learning assistants' office hours**, these are undergraduate students who have previously completed programming and circuit-related courses, and you might find personalized tutor-style help in these office hours. You are also welcome to join any and all instructor's office hours as well.
6. If you ever need an **extension**, contact your lab TA (in-person is the quickest way to get the extension). All legitimate extension requests will be granted for lab assignments. The TA can decide on the length of extension based on individual circumstances. Late work is still eligible for 50% partial credit (no matter how late). You can also use extra credit activities to make up for lost points.
7. **Academic honesty:** You are expected to write your own code and most importantly you must understand all parts of your submissions. Asking for help in understanding a concept, learning from discussions online, and asking ChatGPT for explanations are all OK — but you must write your own code after you have understood the concepts. Remember that learning computer programming will enhance your future education and career, so copying code that you did not write will impede your learning and progress. With advanced code similarity checkers, it is easy for us to find out instances when you did not write your own code, so if such incidents occur you may receive a note from the course staff. Please refer to the full university academic honesty policy PDF.
8. **Class Conduct and Community:** Remember that contributions from each of us can help in building a respectful, courteous, and intellectually stimulating class environment. Language or behavior that prevents any student from participating fully in class is not acceptable. It is important to remain open to each other's thinking and engage in rigorous, challenging discussion about issues of shared concern. This is distinct from participating in oppressive behaviors — racism, sexism, homophobia, transphobia, and ableism, that are designed to keep people out of conversations, not bring them in. Please refer to the full student conduct policy PDF.
9. UC Merced is committed to providing an **equal opportunity environment** for all students and employees that remains free of all forms of discrimination, harassment, and exploitation. Discrimination and harassment based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or University policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. If you witness or experience any form of harassment, please seek support and guidance. For more information, please visit https://ophd.ucmerced.edu/
10. **Emailing/Direct messaging on CatCourses:** If the syllabus does not answer your question, please reach out to the TA in the lab section or the office hour. If your question needs urgent attention, you can send an email to the TA or the instructor (make your best

judgement on who might answer your question best and quickest). Please **mention the course number in the subject** of any email you send.

Assessment structure:

1. All in-class activities and quizzes are graded for participation – 10% of your total grade.
2. All lab check-outs are graded for correct completion (you have unlimited attempts to do these). Two lowest scoring lab check-outs will be dropped from final grade computation. Gradescope autograders will provide instant feedback on errors and mistakes in your code – 10% of your total grade.
3. All lab assignments require more independent effort than the reading assignments and are graded for correctness – 30%. Two lowest scoring lab assignments will be dropped from the final grade computation.
4. Your summative assessment consists of two projects. Each project is worth 25% of your final grade with a total of 10 project milestones (5 milestone assignments for each project).
5. Project grading structure: 30% from milestones (1 lowest scoring milestone will be dropped) + 40% code submission + 30% demonstration. (0 on project if it is not demonstrated).
6. You can use extra credit activities to earn a maximum of 5% toward your total grade.
7. Grading scheme according to final score (out of 105, because of 5 extra credit points):
    1. Score in [96, 105] is A+, [92, 96) is A, and [88, 92) is A-
    2. Score in [84, 88) is B+, [80, 84) is B, and [75, 80) is B-
    3. Score in [70, 75) is C+, [66, 70) is C, and [62, 66) is C-
    4. Score in [58, 62) is D+, [54, 58) is D, and [50, 54) is D-

Midterm project guidelines

All students must work on an independent Python project with clear electrical engineering motivation or a prototype that informs the data for the project. This project must demonstrate the following five key elements of Python programming that you learn in EE 021. An ideal project that grades 100% will

1. Control the program flow with branching and loops
2. Uses correct data structures for optimal computations, as relevant for electrical engineering.
3. Uses functions for modular code
4. Loads data from real electrical circuits stored as files (or data available on the internet), writes outputs to files (if needed), and visualizes outputs in graphs (as applicable)
5. Documents the flow of logic with comments, docstrings, and user-friendly messages

What project can I work on for the midterm?

A few example project ideas are listed on this Google document but you are more than welcome (in fact, encouraged!) to come up with your own ideas that satisfy the project requirements (and ensure that it appears on the Google document by working on midterm project milestone 02). You

can also pick up a particular extra credit idea from one of the lab assignments and expand on them for your midterm project.

Final project guidelines

All students must work on a final project out of four possible options: Advanced Python, MATLAB, C++, or Assembly programming in MIPS. This project must demonstrate the following five key elements of programming that you learn in EE 021.

The final project problem statement PDFs are given below:

- Advanced Python: Image processing
- MATLAB: GUI for "e-view"
- C++: I/O interface for "e-view"
- Assembly: Algorithms and memory

What project can I work on for the final?

Out of the four choices above, you can choose one that suits your planned major the most. A handy graphical interface is available on the final project milestone 01 to help you make the choice.

How can I get full points on the final project?

Check out the rubric posted on the Final project code and report and the Final project demonstration assignment to analyze and plan out your project so that you can get the desired grade in this course.

Schedule for help sessions before the finals

Date	Time	Person	Primary	Additional	Location
Thursday, Dec 5	11:30 AM - 1:20 PM	Shariful	MATLAB	Python, C++	ADMIN 364
Thursday, Dec 5	12:00 PM - 1:00 PM	Kavin	Assembly	C++, Python	SE2 Lobby
Thursday, Dec 5	5:30 PM - 7:20 PM	Shariful	MATLAB	Python, C++	ADMIN 350
Friday, Dec 6	1:00 PM - 2:00 PM	Chandra	Grading Q&A	C++, Python	SE2 Lobby
Friday, Dec 6	1:00 PM - 2:00 PM	Jayesh	Python	C++	SE2 Lobby
Friday, Dec 6	2:00 PM - 3:00 PM	Max	Python	C++	SE2 Lobby
Monday, Dec 9	4:00 PM - 5:00 PM	Kavin	Assembly	C++, Python	SE2 390
Monday, Dec 9	5:45 PM - 6:45 PM	Chandra	Grading Q&A	C++, Python	SE2 Lobby
Monday, Dec 9	1:00 PM - 2:00 PM	Shariful	Exam Q&A	N/A	SE2 Lobby
Tuesday, Dec 10	1:00 PM - 2:00 PM	Jayesh	Python	C++	SE2 Lobby
Friday, Dec 13	2:00 PM - 3:00 PM	Max	Python	C++	SE2 Lobby

Course timeline

Date	Topics	Related assignments and due dates
08/28	Week 0: Intro to Computing	Syllabus quiz, AI survey
09/02	Labor day holiday	--
09/04	Week 1: Circuit breadboards, Python inputs, and outputs	Lab 1 checkout
09/09	Week 2 Lecture 1: Decision-making in computing with branching	Lab 1 due. Midterm project milestone #1 due
09/11	Week 2 Lecture 2: Complex decision-making flowcharts	Lab 2 checkout
09/16	Week 3 Lecture 1: Iterative computing with while loops	Lab 2 due. Midterm project milestone #2 due
09/18	Week 3 Lecture 2: Iterative computing with loops	Lab 3 checkout
09/23	Week 4 Lecture 1: Iterative computing with loops; more circuits practice.	Lab 3 due. Midterm project milestone #3 due
09/25	Week 4 Lecture 2: More practice, generating data, and plotting.	Lab 4 checkout
09/30	Week 5 Lecture 1: Files and for loops	Lab 4 due. Midterm project milestone #4 due
10/02	Week 5 Lecture 2: Data structures: strings and lists with for loops	Lab 5 checkout
10/07	Week 6 Lecture 1: Data structures: strings, lists, and dictionaries	Lab 5 due. Midterm project milestone #5 due
10/09	Week 6 Lecture 2: Dictionaries	Lab 6 checkout
10/14	Week 7 Lecture 1: Functions and modularity	Lab 6 due. Midterm project demo #1
10/16	Week 7 Lecture 2: Guest lecture	Midterm project demo #1
10/21	Week 8 Lecture 1: Project discussions	No lab due. Midterm project demo #2
10/23	Week 8 Lecture 2: Project discussions, open-source tools.	Midterm project demo #2
10/28	Week 9 Lecture 1: Object-oriented programming	Midterm project due;
10/30	Week 9 Lecture 2: Object-oriented programming	Lab 7 checkout;

11/04	Week 10 Lecture 1: Abstractions and skills mapping	Lab 7 due. Final project milestone #1 due
11/06	Week 10 Lecture 2: Skill translation: C++	Lab 8 checkout
11/11	Veterans Day Holiday	Lab 8 due on 11/12. Final project milestone #2 due on 11/14
11/13	Week 11: Skill translation: C++	Lab 9 checkout
11/18	Week 12 Lecture 1: Skill translation: MATLAB	Lab 9 due;
11/20	Week 12 Lecture 2: Skill translation: MATLAB	Lab 10 checkout
11/25	Week 13: Graphical user interfaces	Final project milestone #3 due
11/27	Thanksgiving holiday	
12/02	Week 14 Lecture 1: Pointers	Lab 10 due. Final project milestone #4 due
12/04	Week 14 Lecture 2: Finishing the final project	Final project demo practice
12/09	Week 15: Class review/Project discussion	Final project milestone #5 due. Final project demo in-lab exam
12/11	No class	Final project demo in-lab exam. Final projects due on 11/13
12/23	--	--