Syllabus

ENGR 21 Fall 2026, Swarthmore College

Emad Masroor

Table of contents

What is E21 about?

This course is called Computer Engineering Fundamentals, and its goal is to teach you *how to use a computer to do things that engineers need to do*. As part of the core Engineering curriculum at Swarthmore, E21 is not just for computer engineers, but is a broad introduction to the fundamentals of engineering with computers. This course will teach you skills that are applicable to a broad variety of tasks encountered by engineers of *all* kinds, including mechanical, electrical, civil, computer, chemical, aerospace, biomedical, etc.

E21 is **not** a course about how computers work; we don't take computers apart and we don't learn how to make computers.

There are three major thrusts of this course: an engineering-oriented **introduction to programming**, an introduction to microcontrollers and **embedded systems**, and an introduction to **numerical methods**.

Introduction to programming

By the time you take this course, we expect that you will already have used MATLAB in E6. However, we will start fresh using a new, high-level, open-source programming language: Python. This language is industry-standard and extremely versatile, and you don't need to buy anything to use it. Coupled with the extensive ecosystem of packages that you'll dip your toes in, you'll find that Python is a very useful tool in your engineering toolbox.

Embedded systems

Embedded systems are small-scale computers — also called microcontrollers — embedded on a physical device that allow you to embed logic into physical devices. In this course, we'll make use of the Circuit Playground Express to learn how to write programs for embedded systems.

Numerical methods

This course will also introduce you to numerical methods for engineering. Given an equation or a system of equations, how do you teach a computer to find the solution to those equations? You will learn how to write algorithms for solving equations that often arise in an engineering context using Python's numpy package.

The place of E21 in the curriculum

E21 is a core course for the ENGR major at Swarthmore, and serves as a gateway course for several computer engineering courses, including Mobile Robotics, Computer Vision, and Embedded Systems.

E21 was first taught in the fall of 2024, and it has been taught every fall since then. As a core course, it combines two previously-taught courses: E15, Fundmanetals of Digital and Embedded Systems, and E19, Numerical Methods for Engineering Applications. Students should take this course in sophomore year, or latest by junior year.

How will E21 be run?

Summary

Components

- Two lectures each week. Attendance at these lectures is required.
- A homework assignment each week, due on Thursdays at midnight. This will sometimes consist of two parts: a written part and a coding-based part.
- Five/six laboratory assignments during the semester, approximately once every two weeks
- Six tests, held approximately every two weeks. No midterm or final exams.
- A final project.

Support

- Wizard Sessions on Tuesday evenings, 7:00 to 9:00 PM in Singer 221.
- Instructor-led Problem Sessions Monday 4:30 to 6:30 PM in Singer 221.
- Appointments with Prof. Masroor on Mondays, Tuesdays and Thursdays.
- Open office hours by Prof. Masroor 9 — 10:30 AM on Wednesdays.

Lectures

This course will be conducted in the form of in-person lectures during the regularly-scheduled meeting time. Lectures will not be recorded and remote participation is not possible. Students are expected to attend every lecture, and to take their own notes. Although the instructor will post lecture slides from class to the website, these do not replace the need for students to actively engage with the material, ideally by taking their own notes.

There are at least three different modes of learning that you will spend your classroom time in. These are:

1. Actively listening to a lecture, ideally while taking notes;
2. Discussing in-class questions with your peers, TAs, or the instructor;
3. Using a personal laptop or other computing device to write and execute code during an in-class activity.

Be prepared to switch between these modes of learning as the situation requires it.

Office Hours

You are welcome to come talk to the instructor individually at any time you see the door to Singer 112 open. In addition, you can book an appointment for office hours on Mondays, Tuesdays and Thursdays at various times. Book an appointment here.

Office Hours are Wednesdays 9 to 10:30 and are open, no appointments.

Homework

Homework assignments are an integral part of this course, as with all engineering courses. They will include both programming (in which case you submit code files) and written work (in which case you submit PDF files of typed or handwritten text). All submissions will be electronic. The instructor has designed the homework problems to complement and expand what is covered in lecture. Since this is primarily a course about using computers, there's only so much that can be taught in a classroom setting. Much of the learning in E21 happens *by doing*, whether on your own or while working with others.

Collaboration with your peers is encouraged, but everything you submit must be your own work. A reasonable expectation for the amount of time that you should expect homework to take is ~ 6 hours per week.

Homework will be assigned approximately every week, and will typically be due by midnight on the Thursday following the week in which it was assigned. You can generally expect HW **n** to cover the material from week **n**, and to be due in week **n+1**.

Homework will be submitted on Moodle using Gradescope.

Tests

There will be six tests in this class. There are no midterms or final exams.

The purpose of holding six tests instead of one or two midterms is so that you have low-stakes opportunities to demonstrate your mastery of the course material at regular intervals. In general, you can expect these tests to be independent of each other, as opposed to being cumulative.

These tests will be held on Tuesdays during the first half of class time (typically for 25-30 minutes), and will be closed-book, closed-notes and closed-computer. The tests will usually be incremental rather than cumulative, and will only cover the material from approximately two weeks prior to each test. Typically, Test **n** will cover the material from week **2n** and week **2n-1**, and will be held on week **2n+2**; for example, test 1 will cover weeks 1 and 2 and will be held on the Tuesday of week 4; test 2 will cover weeks 3 and 4 and will be held on the Thursday of week 6, etc.

Labs

Labs will be held on alternate weeks.

Final Project

Instead of an exam, E21 will conclude with a final project. This will be an opportunity to use what you've learned in the class to design and build a system that uses Computer Engineering Fundamentals in creative ways. See a list of past projects here

Course at a glance

Field	Info
Course Number	ENGR 021
Course title	Computer Engineering Fundamentals Analysis
Type	Core ENGR course
Lab	Meets biweekly
Enrollment	60+
Communication	Moodle, this website, Gradescope, EdStem
Prerequisites	ENGR 006, MATH 025, ENGR 017
Lecture Instructor	Emad Masroor
Lab Instructor	Matt Zucker
Lectures	TR 8:30 to 9:45 SCI 101
Problem Session	Monday 4:30 to 6:30 PM Singer 221
Wizard Session	Tue 7:00 to 9:00 PM Singer 221
Midterm Exam	None
Final Exam	None
Tests	Six Tests
Optional Textbook	*Programming with MicroPython*, Tollervey, Nicholas H.
Optional Textbook	*Numerical Methods In Engineering with Python 3*, Kiusalaas, Jaan

Catalog Description

This course will introduce students to foundational topics in computer engineering. Key topics include digital representation of data, gates and truth table, and basics of computer organization. Students will gain proficiency coding in Python in both the embedded and traditional computing contexts. Students will also gain exposure to key methods for computational modeling of engineered systems and numerical solution of problems in engineering.

Schedule

This table is best viewed online.

Week	Class		HW Due	Topic	Test	Lab
1	1	Tue,Sep 1		Intro to E21 & Python		
	2	Thu,Sep 3	HW 0	Programming + Circuit Python Express		
2	3	Tue,Sep 8		Base systems; Analog/Digital data		
	4	Thu,Sep 10	HW 1	More on programming basics		Lab 1 starts
3	5	Tue,Sep 15		Quantifying errors		
	6	Thu,Sep 17	HW 2	Errors, Precision & Accuracy		
4	7	Tue,Sep 22		Floating Point	Test 1 on HW 1 & 2	
	8	Thu,Sep 24	HW 3	Floating Point		Lab 2 starts
5	9	Tue,Sep 29		Intro to numpy; writing data to CPX		
	10	Thu,Oct 1	HW 4	Data visualization		
6	11	Tue,Oct 6		State Machines	Test 2 on HW 3 & 4	
	12	Thu,Oct 8	HW 5	State Machines		Lab 3 starts
		Fall Break		Fall Break		
7	13	Tue,Oct 20		Root-finding		
	14	Thu,Oct 22	HW 6	Root-finding		
8	15	Tue,Oct 27		Linear systems	Test 3 on HW 5 & 6	
	16	Thu,Oct 29	HW 7	Linear systems (cover Gaussian elimination)		Lab 4 starts

Week	Class		HW Due	Topic	Test	Lab
9	17	Tue,Nov 3		Linear systems		
	18	Thu,Nov 5	HW 8	Linear systems		
10	19	Tue,Nov 10		Optimization / Initial Value Problems	Test 4 on HW 7 & 8	Intro Project
	20	Thu,Nov 12	HW 9	Optimization / Initial Value Problems		Lab 5 starts
11	21	Tue,Nov 17		Optimization / Initial Value Problems		
	22	Thu,Nov 19	HW 10	Optimization / Initial Value Problems		
12	23	Tue,Nov 24		Curve-fitting & interpolation	Test 5 on HW 9 & 10	
	0	Thanks-giving	HW 11			Project
13	24	Tue,Dec 1		Curve-fitting & interpolation		
	25	Thu,Dec 3	HW 12	Work on final project		
14	26	Tue,Dec 8		Work on final project	Test 6 on HW 11 & 12	

Notes

* Prof. Masroor will be away for a conference † Content of HW 12 will be announced later

Teaching Team

Name	Role	Office Hours
Emad Masroor	Lecture Instructor	Wed 9 to 10:30 & by appt.
Matt Zucker	Lab Instructor	
Visnuk Rith	Course Wizard	
Samchan Lee	Course Wizard	
Uneeb Hyder	Course Wizard	
Hiba Aby	Grader	
Huyen Nguyen	Grader	
Ellie Fermo	Grader	
Kevin Sisk	Grader	

Grades

In this course, it is possible for you to calculate your overall numerical score by using the following table of weights. Your final score in the course is a weighted average of the components. Each homework assignment will be weighted equally.

Components of Grade

Component	Grade
Participation	5%
Homework	20%
Tests	45%
Labs	15%
Final Project	15%

Grade Thresholds

This class uses an absolute threshold for grades. Your grade in E21 is not based on a comparison with your peers' performance, but is instead based on an absolute standard set by the instructor. Numerical threhsolds will be used to convert numeric scores to letter grades according to the Swarthmore College grade definitions.

> **Definitions of Letter Grades**
>
> From the Swarthmore College catalog: A means excellent work; B, good work; C, satisfactory work; D, passing but below the average required for graduation; and NC (no credit), uncompleted or unsatisfactory work.

The instructor will assign +/- qualifiers to the letter grades, approximately equally spaced across the corresponding letter grades, i.e., 83 is the minimum for a B and 87 the minimum for a B+.

> **Adjustment of thresholds**
>
> One or more of the numeric thresholds set out in this document may be adjusted downward in your favor (i.e., by making it easier to get a certain letter grade) and will not be adjusted upward. Such changes are made to bring letter grades in alignment with the College's definitions.

Letter	Numerical score
A range	90-100
B range	80-90
C range	70-80
D range	60-70
NC	Below 60

Policies

AI Policy

This is a course for humans by humans. This means that

- AI will not be used by the instructor to create lecture slides, write homework assignments, or set tests. It will not be used to grade any student work.
- Students are likewise expected to complete the course requirements without using AI.

Statement of Instructor's Guiding Principles

The following comments were given by the instructor to the Swarthmore Phoenix in November 2025.

> I believe that the wide availability of generative "artificial intelligence" is an impediment to student learning. This is not to say that AI tools are not useful, but simply to say that, on balance, they are harmful in an educational context and likely to lead to a serious deterioration of students' ability to write well, think critically, and read deeply. By short-circuiting the difficult process of learning, these AI tools give us the illusion of knowledge while in actuality being a simulacrum of the real thing. After all, if you only know how to do something with the help of a chat bot, do you really know how to do it? And, perhaps more relevant for students entering the job market, why would anyone employ you for a "skill" that anyone else with an internet connection could just as well claim to have?
>
> The path from ignorance to knowledge is not an easy one. It is challenging, and struggling against that challenge is, pretty much, the entire point of the educational enterprise. Forgive me for having a luddite opinion here, but I think it is quite possible that some technology is actually bad for society, and that innovation can be regressive instead of progressive. A mass plagiarism machine that can do students' homework for them without their having to lift a finger, compose their essays, write their code, make their presentations, summarize their readings, and even answer interview questions in real-time is, in fact, just as bad as it sounds.
>
> While it is true that these tools are proliferating in many professions — leading some to contend that colleges must "prepare their students for the AI age" — I believe that faculty at a liberal arts college should exercise discernment in their desire to keep abreast of this latest fad. To students, I would say that no matter how much the AI maximalists would like to tell you otherwise, there will never be any substitute for thinking, reading, and writing. These three activities are essential to the formation of young people and have been the bedrock of education in civilized societies for thousands of years. To the extent that a new tool promises to alleviate the burden of having to think, to read, or to write, that tool offers us only a devil's bargain that will leave us poorer of mind and spirit and will rob our students of a true education.

Prohibited use of AI

- You are not **entirely** prohibited from asking Large Language Models questions (in your own words) related to this course. However:
  - Many of the assignments in this class require you to complete a task by writing code that accomplishes this task. The following are some ways in which you could complete the task:
    1. You could copy a friend's code and turn it in.
    2. You could copy a solution you found on the internet.
    3. You could Google "How do I x", read the approach suggested by the first search result, and then implement the approach in your own words.
    4. You could ask a LLM "How do I x", and then implement the approach suggested by ChatGPT/Claude/Gemini.
  - The common thread between these four approaches is that by adopting them you bypass the most important element of taking an Engineering class: **thinking** about the assigned problems and coming up with **your own solutions** to them.
  - All four options above are unacceptable for this class. You should be able to complete all assignments in an AI-free and even internet-free environment.
- You may not upload assignment instructions, lecture slides, or other course material to an AI service.
- You are prohibited from using Large Language Models to produce anything (text, code, diagrams, etc) that you turn in.

Violations of this policy will be treated as cases of academic misconduct.

Isn't most code written by AI now ?

E21 is a programming-heavy course. You may be aware of news reports that claim that '75% of all code is now written by AI', or some other large number. By some accounts, enormous quantities of code are now being generated at speeds faster than humans can review it, and you may be wondering why the AI policy in this course is seemingly out of step with these developments in industry.

- The prohibition on AI in E21 is akin to the calculator-free classrooms of elementary school mathematics. It is important to learn programming, *especially* if you go on to write programs for a living with or without AI assistance.
- A successfully completed college course requires you to produce a body of work that represents your own intellectual output. The result of a prompt given to an AI tool is **not** your intellectual work.

Attendance at Lecture

- is required.
- Please be respectful of everyone else's time by being punctual.
- Cell phone use is not permitted in class. If you have an urgent matter to attend to that requires you to use your phone, you may leave the room discreetly and return when you are able to.

- You are encouraged to bring a personal (laptop) computer to class. However, these should be used during in-class programming activities and should be put away (i.e., screen closed) otherwise.

Homework Policies

- Late work will not be accepted for grading.
- Exceptions to the above may be granted for extenuating circumstances, such as sickness, family emergencies, and other circumstances beyond one's control. These exceptions will be at the discretion of the instructor.
- Each student can get **one 'free' extension** during the semester, of one week's duration, with no penalty. This is a no-questions-asked extension and will be deemed automatically granted when you request before the deadline.
- Both types of extensions must be requested via Google Forms here.

Test Policies

- A test can be rescheduled in extraordinary circumstances for reasons of medical emergency, family illness, etc. Rescheduled tests will typically occur in the form of an oral exam in person.
- A test missed without prior notice will result in a score of zero.
- Grades on tests can be disputed, but only within one week after graded tests are returned.

Lab Policies

- Attendance at lab is required for the full duration (1:15-4:00 PM).
- Please be respectful of everyone else's time by being punctual.