

EECS 490 Fall 2019

Syllabus

Instructors

Amir Kamil akamil@umich.edu

Staff

Noa Levi nlevi@umich.edu
Gabriel Matute gmatute@umich.edu
Mitchell Morrell morrelmi@umich.edu
Makarand Parigi parigim@umich.edu

Contact

Please direct technical questions to our Piazza forums. For administrative questions, you can reach the course staff at eeecs490staff@umich.edu.

Schedule

Tue, Thu	12:00 pm - 1:30 pm	1013 Dow	Lecture
----------	--------------------	----------	---------

For office hours, please refer to the calendar on the course website. In order to ensure fair access to all students, we will not provide any help to students outside of scheduled class meetings, office hours, and Piazza.

Textbook

Optional: *Programming Languages: Principles and Paradigms*, by Gabbrielli and Martini.

Our textbook is available in print form and DRM-free electronic from [here](#).

Course Notes

A set of course notes will be available for each topic on the Google Drive. These notes are **required** reading, and you will be responsible for any material covered in the notes, unless it is explicitly marked as optional.

Overview

This course covers the fundamental concepts in programming languages. While we will be using several languages, the purpose of this course is not to learn different languages. Instead, it is to learn the concepts that will allow students to learn a new language quickly and to make better use of the programming constructs that are provided by a programming language. To analogize with spoken languages, this course is more akin to linguistics rather than specific languages.

The official course description is a reasonable description of what the course covers:

Fundamental concepts in programming languages. Course covers different programming languages including functional, imperative, object-oriented, and logic programming languages; different programming language features for naming, control flow, memory management, concurrency, and modularity; as well as methodologies, techniques, and tools for writing correct and maintainable programs.

We will also cover basic theory of programming languages, such as grammars and type systems, as well as advanced programming techniques such as generic programming and code generation.

Topics

- Foundations
 1. Names and environments
 2. Control flow
 3. Memory management
 4. Grammars

- Functional programming
 1. Recursion
 2. Higher-order functions
 3. Continuations

- Theory
 1. Lambda calculus
 2. Operational semantics
 3. Type systems

- Data abstraction
 1. Object-oriented programming
 2. Inheritance and polymorphism
 3. Static and dynamic typing
 4. Generics
 5. Modules
- Declarative programming
 1. Logic programming and unification
 2. Constraints and dependencies
- Metaprogramming
 1. Macros
 2. Code generation
 3. Template metaprogramming
- Concurrent programming
 1. Parallel programming

We will explore these topics primarily in Python, Scheme, and C++, though we may discuss other languages as well.

Prerequisites

The prerequisite for EECS 490 is EECS 281 or graduate standing. We recommend having some experience in Python, as most of the assignments will be in Python 3.

Website (eecs490.org)

The first place to go for any course materials or resources is our course website at eecs490.org (you'll need to sign in with your @umich.edu account). All course materials and assignments are made available there, and are considered **required** reading. A detailed schedule, including lecture topics, assignment due dates and exam dates, is also available there. Other resources such as Canvas, Piazza, Gradescope, and the autograder are linked from the site.

Minimal additional information can be found on the course Canvas site. Make sure you are set up to receive course announcements on Canvas, since we may post critical information there.

Forum

We will be using Piazza to host a course forum. You are required to read this regularly; it is the venue we will use for important course announcements and project clarifications. In addition, it will be a significant source of help and hints on the assignments and projects.

We do not answer technical questions via email. In order to save everyone time, we want all students to have the benefit of seeing each question and its answer, so please use the forum.

Do not post your own solutions, assignment or project code, test cases, or output to the forum. Also, **please search the forum before posting** to avoid questions that have already been asked and answered.

Projects and Homework

Over the course of the semester, we will assign five programming projects. The first is a short project intended to give you some familiarity with Python. The latter four projects are more substantial explorations of concepts in programming languages and compilers. There will also be five homework assignments, consisting of a combination of short programming exercises as well as written questions.

Corrections and Clarifications

We will post a Canvas announcement in the case of any critical corrections or clarifications. It is your responsibility to ensure you are able to receive Canvas announcements.

Programming Environment

You are free to develop your programs on any platform you like, but you may use only standard features and libraries of the programming environment used for an assignment (e.g. Python 3.7, Scheme R5RS, and C++17).

Programming assignments **must** be submitted to the **autograder** before the deadline. The written portion of homework assignments **must** be submitted to **Gradescope** before the deadline.

The Autograder

We use a web-based autograder for project submissions and the coding portions of homework. You will receive 3 submissions per day for each assignment. Each submission will receive feedback from the results of the public test cases released with the assignment. Public test cases contain incomplete feedback; just because your code passes the public test cases, that does not mean it is 100% correct! After the deadline, the autograder also runs a second set of private test cases that will be used to determine the correctness score for your submission. Your final assignment grade will be the score of the submission that received the combined **best** score on the public and private tests. If multiple submissions share the best score on the tests,

we will grade the last such submission.

Several of the projects will also be graded for good programming practices, both by hand and using automated tools. We will hand grade the same submission that is used to determine your correctness score. Since scores on private test cases are not released until after the deadline, this means that it may not be apparent before the deadline which submission will be hand graded. **You should always use good programming practices**, and doing so will help you avoid losing points because we hand graded a submission you did not expect.

Successfully Completing the Projects

We have found through many years of teaching experience in various courses that the most common reason for poor project performance is **not starting early enough**. Plan to do some work on the project every day and try to have it finished a few days ahead of the due date, because many unexpected problems arise during testing. In addition, the computing sites can become very crowded, making it difficult to get a computer to use, so plan for these things to happen.

The second most common reason for not doing well on the projects is not asking for help when you need it. We offer help in office hours and on the class forums. When you come to office hours, please be ready to provide access to your code, preferably electronic. Another good way to get help is to post a question to the course forum. Remember, if you find that you are stuck on a piece of your project for an undue amount of time, please see us!

An important skill in programming is to be able to test and debug your programs independently. As such, we will not debug your programs for you. Instead, we will help you try to figure out how to test and debug your program yourself. We will also ask you to demonstrate what testing and debugging techniques you have already tried, and what the results were, before providing any advice.

Finally, use a revision control system such as git, and be sure to commit and push your work to your repository often so that you don't lose your work. If you somehow lose your work, it is your responsibility.

Project Evaluation and Grading

Each project will be evaluated for behavioral correctness, adherence to course principles, and good programming practices. Some projects may also be evaluated on the quality of test cases you supply.

A program is correct if it behaves as specified in the project handout, for example, by generating the correct output. A program adheres to course principles when it uses techniques taught in lecture and asked for in the specification. Finally, a program should be readable by other programmers. We recommend adhering to common style guidelines for each programming

environment, such as PEP 8 for Python.

Project Checkpoints

Projects 2 - 5 will have checkpoints that require you to meet a specific correctness threshold in advance of the final deadline. Refer to the project specifications for details.

Due Dates and Extensions

Assignments turned in after the exact time and date on the assignment will receive a zero. We do not generally offer extensions. For example, we do not offer extensions due to crowded computing sites, long queue times or slow response times on the autograder, internet access problems, accidental erasure or loss of files, or outside conflicting commitments.

We will consider extension requests made in person and at least two weeks in advance, for example, for religious holidays or planned medical procedures.

Additionally, we will consider requests for extensions due to documented, unanticipated medical or personal emergencies. If you can't see the instructor in advance due to the emergency, then contact them as soon as you possibly can. In all cases, we require written proof of the emergency.

Academic Integrity

You may give or receive help on any of the concepts covered in lecture, discussion, reading assignments, or the textbook. You are allowed to consult with other students in the class to help you understand the assignment specification (the definition of the problem).

You must complete Project 1 alone. You may complete homework assignments and Projects 2 - 5 either alone or with a partner. **For those retaking the course: if you submitted a project in a previous term, you may not partner on that same project this term.** All programming assignments in this course are to be done by you or your partnership.

You may not collaborate in any way with people outside your partnership when constructing your solution; your partnership working alone must generate the solution to a programming assignment and must submit your code for grading together (i.e. your partnership may only submit one version of your code for grading). You are not allowed to work out the programming details of the problems with anyone outside your own partnership or to collaborate to the extent that your programs are identifiably similar. You may not derive your solution in any way from other solutions. (If you are retaking the course, you may reuse your own code.) **You may not share code outside of your partnership, whether during or after the semester, including making it publicly available in any form (e.g. a public GitHub repository).** **You may not share test cases outside of your partnership as we consider your test cases part of your solution.** If you have any questions as to what constitutes unacceptable collaboration, please talk to the instructor right away. You are expected to exercise reasonable precautions in protecting your

own work. Do not let other students borrow your account or computer. Ensure that the computers you use to access project code are password protected. Do not leave your program in a publicly accessible directory, neither during the semester, nor after. Take care when discarding printouts. You are still responsible for following these rules even after finishing the course.

You may not submit any code that violates the rules and regulations in the assignment specification, or that attempts to subvert the grading system. The latter includes hard-coding solutions to public test cases, using runtime introspection to obtain information about private test cases, and so on. Code that violates these rules will be summarily given a zero.

You may **not** submit an iClicker answer for someone else.

We report suspected violations to the Engineering Honor Council. To identify violations, we use both manual inspection and automated software to compare present solutions with each other and with prior solutions. The Honor Council determines whether a violation of academic standards has occurred, as well as any sanctions. Read the Honor Code for detailed definitions of cheating, plagiarism, and other forms of academic misconduct.

Rules for Partnerships

Working in a partnership is optional. For homework assignments and Projects 2 - 5, you may work either alone or in a partnership. The following rules apply to those who choose to work in a partnership.

If you work in a partnership, you and your partner will submit one assignment together. You will need to register your partnership for a project on the autograder before making the first submission for the assignment. You cannot change partners in the middle of one assignment. You may change partners only after a project is completed and submitted. **You must work in the same partnership for Projects 4 and 5. If you work alone on Project 4, you must also work alone on Project 5.**

You and your partner must collaborate on all aspects of your solution. You may **not** split up the project and work on separate parts individually. **Both partners will be held equally responsible for anything submitted on behalf of the partnership.**

DOs

- Do READ THESE RULES CAREFULLY before programming with another student. You must follow these rules, or risk being investigated for an Honor Code Violation.
- Do choose a partner from the current semester of this course.
- Do register your partnership on the autograder and/or Gradescope before making a submission for an assignment.
- Do submit one copy of the assignment together.

DON'Ts

- Do not program with someone without understanding these rules.
- Do not partner on an assignment with someone who has already made progress on the problem.
- Do not share code, **test cases**, or written solutions with anyone other than your partner, or a staff member.
- Do not split the work in half. Both partners must work on all parts of the solution.
- Do not partner with anyone who is not currently enrolled in the course.
- **Do not partner with anyone on a project that you have submitted in a past term of EECS 490.**

Discussion Sections

In addition to lectures, the course has discussion sections scheduled for each week. Discussion sections will cover more detailed examples and exercises than lectures, as well as provide help on assignments.

Grading and Exams

Your final grade is based on scores from iClicker exercises, homework assignments, programming projects, survey participation, and two exams. The tentative point distribution is included in the following table. It is not likely that this will change, but circumstances might occur that would make changes necessary, at the discretion of the instructor.

Assignments	Without iClicker	With iClicker
In-lecture iClicker exercises	0%	10%
Homework assignments	15%	13.5%
Programming projects (4% or 3.6% for P1, 9% or 8.1% for each of P2-P5)	40%	36%
Midterm exam	20%	18%
Final exam	24%	21.6%
Participation in course surveys, etc.	1%	0.9%

There are no letter grades for individual projects or exams. The final course letter grade is based on the total weighted score earned. To pass EECS 490, your weighted average project score must be a passing score, and your weighted average exam score must be a passing score. If you score 60% overall, and your exam average is above 50%, and your project average is above 60%, you will pass the course with a C or better. Final grades will be assigned based on the distribution of earned scores. We expect the median student to receive a grade in

the mid B range, and students within one standard deviation of average can generally expect to pass. We may adjust our distribution if this term's group does particularly well or particularly poorly. In particular, if everyone does great work, we'll assign everyone a great grade! Please note, we do not offer the opportunity for "make-up" or extra credit work to improve your grade.

We will compute your total weighted score in two ways, and your final score will be the **maximum** of the two. The first is without any credit for iClicker exercises, and the second is with iClicker exercises as 10% of your grade and the remaining assignments constituting 90% of your grade. Refer to the table above for the point distributions for the two cases.

iClicker Exercises

Many lectures throughout the semester will include iClicker exercises. Your iClicker score will be computed out of 15 points. You will receive 1 point for each lecture you answer at least 50% of the iClicker questions for that day. There are more than 15 lectures with iClicker exercises, so you have many opportunities to earn all of the 15 possible iClicker points. Because of this, we will not provide accommodations for lost clickers or dead batteries. You are responsible for registering your clicker on Canvas as well as updating the registration and notifying eeecs490staff@umich.edu should you replace your clicker during the semester.

You may not submit an iClicker answer for someone else. Doing so is a violation of the Honor Code and will be reported to the Engineering Honor Council.

Exams

We expect you to take both exams at the scheduled times. If you miss an exam, and a medical or personal emergency is not involved, you will receive a zero for that exam. If you anticipate an exam in another course or a religious holiday that conflicts with our exam time, you must notify the instructor **within the first three weeks of the term**. The exam dates are given at the beginning of the term so that you can avoid scheduling job interviews or other commitments on exam days; hence job interviews, etc. are not considered valid reasons for missing an exam.

Midterm: Thursday, 24 October 2019, 12noon - 1:30pm

Final: Thursday, 19 December 2019, 8am - 10am

Regrade Requests

While we work hard to grade accurately, we sometimes make mistakes. If you believe we graded an assignment of yours incorrectly, you can submit a regrade request before the announced deadline for regrades for that assignment. We reserve the right to regrade your entire assignment, which can cause your grade can go up, but it can also go down. Regrade requests should only be made if you feel a grading error occurred - not if you are unsatisfied with the grading criteria.

Please refer to course announcements for the regrade procedure and deadline for individual assignments and exams.

Tips for Doing Well in the Class

You will maximize your grade, and learn a lot at the same time, if you:

- Attend all lectures and discussion sections (note that many tips and hints about projects are given during class!)
- Read the assigned readings (textbooks, course notes, forum, web pages)
- Hand in your work on time (even if a program does not work, turn in whatever you have done)
- Start the programming projects early and come for help as soon as you need it
- Follow the program specifications carefully

Accommodations for Students with Disabilities

If you think you need an accommodation for a disability, please let your instructor know **within the first three weeks of the term**. Some aspects of this course may be modified to facilitate your participation and progress. As soon as you make us aware of your needs, we can work with the Services for Students with Disabilities (SSD) office to help us determine appropriate academic accommodations. SSD (734-763-3000; <http://ssd.umich.edu>) typically recommends accommodations through a Verified Individualized Services and Accommodations (VISA) form. Any information you provide is private and confidential and will be treated as such.

Commitment to Equal Opportunity

As indicated in the [General Standards of Conduct for Engineering Students](#), we are committed to a policy of equal opportunity for all persons and do not discriminate on the basis of race, color, national origin, age, marital status, sex, sexual orientation, gender identity, gender expression, disability, religion, height, weight, or veteran status. Please feel free to contact us with any problem, concern, or suggestion. We ask that all students treat each other with respect.

Students' Mental Health and Well-being

University of Michigan is committed to advancing the mental health and well-being of its students. If you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. For help, contact Counseling and Psychological Services (CAPS) at (734) 764-8312 and <https://caps.umich.edu> during and after hours, on weekends and holidays, or through its counselors physically located in schools on both North and Central Campus. You may also consult University Health Service (UHS) at (734) 764-8320 and <https://www.uhs.umich.edu/mentalhealthsvcs>, or for alcohol or drug concerns, see www.uhs.umich.edu/aodresources. For a listing of other mental health resources available on and off campus, visit: <http://umich.edu/~mhealth>.

Research Disclosure

Your class work might be used for research purposes. For example, we may use anonymized student assignments to design algorithms or build tools to help programmers. Any student who wishes to opt out can contact the course staff (eeecs490staff@umich.edu) to do so at any time up to seven days after final grades have been issued. This has no impact on your grade in any manner.

Right to Revise

The course staff reserve the right to make changes to the syllabus at any time as they see fit. When a revision occurs, it will be announced, and it is your responsibility to be informed of such.