Syllabus

Course Code	CAS CS 392
Course Title	Rust, in Practice and in Theory
Semester	Spring 2025
Instructor	Nathan Mull
Meeting Time	Tuesday and Thursday, 5:00PM-6:15PM
Meeting Location	CDS 364
Midterm Date	March 6
Grade Breakdown	30% Assignments
	40% Final Project
	20% Midterm Exam
	10% Participation

CAS CS 392: Rust, in Practice and in Theory is a course about Rust, a type-safe, memory-safe PL that is becoming a popular alternative to C and C++ in settings where performance and memory usage are major concerns. It's self-described as having "high-level ergonomics" and "low-level control". Practically speaking, this means clear, concise code with fewer memory bugs. Theoretically speaking, this means the use of a rich type system based on the notion of linearity to enforce memory-safety before any code has actually been run.

Despite its popularity, Rust is still daunting to learn, even for experienced programmers. There are several concepts in Rust that don't appear in any other popular languages. And even if you become a proficient Rust programmer, you may not have a deep understanding how Rust works, or why it is a better alternative to other low-level languages.

In this course, we'll spend the first half of the semester learning Rust. We'll spend the second half implementing a subset of Rust *in Rust*. This will help us better understand the details of Rust's type system and borrow checker.

General Information

This course meets 5:00PM-6:15PM Tuesdays and Thursdays in CDS 364. There are no labs/discussions for this course. The course consists of assignments, a midterm exam, a final project, and pre-lecture quizzes.

Disclaimer

Please read this section carefully. This is a new course. It's the first time I'm offering it, it's on a quite modern topic and, to be frank, I'm not sure how well it's going to go. All of this is to say: if you're

looking for a streamlined course experience, then I don't recommend taking this course. This course will require a level of academic independence and maturity that I expect of advanced undergraduate students. There's going to be experimentation, some things that work, some things that don't, some things with bad documentation, some things that will require you to look things up outside of the course material (gasp). But if you stick with the course, I hope it can be a valuable experience for all of us.

Course Structure

Please read this section carefully. This course will be treated as a hybrid of a *flipped classroom* and a *workshop*. Each lecture will have corresponding readings which you will be expected to have read *before* lecture, and which will be required to answer pre-lecture quizzes.

The first part of lecture will be dedicated to a short overview of the day's topic. Each lecture will have corresponding discussion questions. *I want you to think of this part of lecture as closer to your standard humanities course.* It should be a high-participation discussion, with live-coding and board work. It's during this part that you should make sure you're on the same page as everyone else with respect to the material.

The second (primary) part will be dedicated to working with the material from the reading. This will almost always just mean working on the homework assignments. This means that homework assignments will be long, but you will have lots of opportunities to work on it with other students in the course and get help. During this part, you will be encouraged to work in groups of 2-4. Ideally, you'll be able to create a small group that you can work with consistently. Of course, this does not mean that you can cheat (see below for cheating policies) but I want you all to help each other as much as possible.

Lectures are required. You are allowed to miss a handful of lectures due to conflicts, but the structure of this course only works if you want to participate. If you feel like you can't do this (be it because of social anxiety, depression, family conflicts, etc.) then we can have a conversation and see what other options there are. But this does mean that you have to advocate for yourself.

Prerequisites

The formal requirements of this course are *CAS CS 320: Concepts of Programming Languages* and *CAS CS 210: Computer Systems*. You'll have the best time with this course if you took CS320 during the Fall 2024 semester. It is not strictly a requirement, but if you didn't, then you'll have to catch up a bit on the type theory part of the course. We'll have a crash course on this after the spring recess, but it will likely take some time and effort outside of class to master this material. Ideally, we should be able to create groups in which at least one member has this experience.

The dependence on 210 is quite mild. You need to be comfortable with the terminal, and you need to have some sense of low-level memory management. That said, the more versed you are in systems, more you'll get out of learning Rust.

Learning Outcomes

▷ Learn the rudiments of Rust, a notoriously difficult but ultimately rewarding low-level programming language

▷ Better understand the use-cases of Rust, when it works well, when it doesn't

▷ Identify the pitfalls of Rust programming and determine how to avoid them

▷ Learn to read/practice reading academic papers

▷ Learn to read/practice reading formal specifications of programming languages

▷ Implement an interpreter for a subset of Rust in Rust

▷ Prove properties like progress and preservation for a formal specification of Rust (properties which make Rust a "well-designed" programming language)

Evaluation

The grade breakdown for this course is as follows

30%	Assignments (5% each, 2 dropped)
40%	Final Project (10% each part)
20%	Midterm Exam
10%	Participation

Assignments

During the first part of the course, there will be weekly programming assignments, made up either of (1) a collection of programming exercises or (2) a small tool you'll need to build in Rust. There are 8 assignments total. Two of your lowest assignment grades will be dropped.

The assignments will be somewhat long but, as was mentioned above, we will work on them during lecture in a group setting. The hope is that you'll be able to finish most of the assignment during lecture.

Midterm Exam

The midterm exam will be held on Thursday March 6th during class. This is right before the spring recess so plan accordingly. The exam will be on the fundamentals of Rust. I don't expect it too be difficult, it is meant to ensure that you are ready to build a large program in Rust during the second half of the semester.

Final Project

The bulk of the grade in this course will be dedicated to the final project. As mentioned above, the final project is an interpreter for a subset of Rust written in Rust. There will be 4 parts to the project

▷ A parser

▷ An evaluator

▷ A type/borrow checker

▷ An extension of your choosing

You will expected to submit your own work but, as with the assignments, you will also be expected to work in a small group and complete much of the project during lecture. More will be released during after the spring recess.

Participation

This part of your grade will be determined by your lecture attendance and your pre-lecture quizzes.

Course Resources

The resources we'll use for this course are standard. Feel free to reach out directly if you have questions.

All course material will be made available on the course website. Please check it frequently.

A personal note. I'm putting some stricter boundaries on course communication for myself this semester. I will not respond to anything (emails, Piazza posts, etc.) between the hours of 8:00PM–8:00AM.

Piazza

For course communication, we'll use Piazza. Please check it frequently for announcements about deadlines, errata, office hours, etc. And if you have a question about course material, please ask it on Piazza, ideally publicly. I will not respond to emails about course material. A couple basic Piazza policies:

▷ **Don't ask homework questions directly.** Formulate a question which will aid in your understanding, and will hopefully help others as well.

▷ **Don't give homework solutions directly.** The answer-provider is just as culpable as the answer-receiver in cases of cheating.

▷ Piazza is as useful as it is active. I'll be answering questions on Piazza, but don't hesitate to answer questions yourself.

Gradescope

All assignments and project check-ins will be submitted via Gradescope. If you're unfamiliar with Gradescope, see their get-started page for information and tutorials. I will not be setting up autograders this semester, I'll be reading code by hand.[1]

Policies

Collaboration

As mentioned above, there's going to be a lot of group work in this course. My collaboration policy is pretty simple: work in groups, talk about solutions, but **never** send someone your work, copy-paste your work, or let someone copy your work. I will be checking for code-similarity, and notable cases will be investigated. If you collaborate with someone you **must** cite this in your own work. See the next section for more details.

Academic Integrity

Please read the BU's academic conduct code. This is taken very seriously at BU and I take it seriously in the courses I teach.

If you work with others, consult materials found on the Internet, or use an AI assistant, you should cite your sources. This is a useful skill in any setting, and so I recommend being as conservative as possible regarding what you cite. In any assignment, these **citations should appear next to every corresponding problem** (in comments if the submission is code). Some examples:

▷ I discussed problem 1 and 2 with Leah Smith. She helped me understand X and Y aspects of the problem

▷ I saw the stack overflow post *stackoverflow.com/questions/6681284/python-numpy-arrays* which informed my solution

▷ I helped Zihan Guo with problem 4. I told them to try using X

▷ I asked chatGPT "what's the largest eigenvalue of this matrix?" but asked it not to show me how to solve for it

▷ Group-programmed this with Leslie Diggs and then wrote my own solution

It can be tricky determining when a citation is necessary. When in doubt, err on the side of longer, more descriptive citations. And again, **all parties are culpable**. Those who share their work with others (this includes sending files and screen sharing) receive the same attention as those who use that work. Please email me immediately if you are unsure about a particular situation.

---

[1]So, in particular, don't give my bullsh** code. If I feel like you code was generated, it will be investigated for cheating.

The line between collaboration, abuse of online resources, and misconduct has become incredibly blurry, and cases are becoming more difficult to navigate. There are obvious prohibitions:

▷ Don't share your solutions with anyone else

▷ Don't use the solutions of other students

▷ Don't ask homework questions online on sites like Stack Overflow

▷ Don't ask an AI assistant for a solution to a homework question

▷ Don't consult a solution posted online, even to verify your own

It would be unreasonable to believe that these things never happen, or that I can recognize every case that it happens. But I have to depend on the integrity of all my students, which also means that I have to take seriously cases of academic misconduct if they do unfortunately come up. Use your best judgment and remember:

▷ I care about your success in this course. I provide a number of avenues to ask for help, please use them

▷ You will have to answer questions on exams without external aids (and in interviews when you apply for a job)

▷ If you don't know how to start thinking about a problem, *it's okay to ask for pointers on how to start* in office hours and Piazza. I can't give you any part of the answer, but I may be able to put you in the right direction

▷ We have safe-guards (like dropped homework assignments) in the case you are unable to complete an assignment

Disability Statement

If you require disability accommodations, please contact me as soon as possible. You should provide me with the appropriate documentation, available though BU's Disability and Access Services. If there's a policy you see that I'm failing to comply with, please reach out with suggestions. And if you'd like accommodations that are not covered by existing services or policies, feel free to contact me and we can see what we can do. I want everyone to feel able to fully participate in the course.

Diversity Statement

My aim is to present material in a way that respects the diversity of the student body. If I fail to do this, please make me aware. Any suggestions are welcomed and appreciated. I also expect students taking any course of mine to appreciate and respect the unique opportunity they have to participate in a diverse student body.

Sexual Misconduct

Please read BU's Sexual Misconduct Policy and review the entire page for information about talking to someone confidentially about experiences of sexual misconduct, filing a report, and any other relevant information. Above all, you should feel safe, and able to be productive. If this is not the case, please reach out to me or someone else immediately.

The university is legally obligated to investigate reports of sexual misconduct, and therefore it cannot guarantee the confidentiality of a report, but it will consider a request for confidentiality and respect it to the extent possible. I am also required to report incidents of sexual misconduct and **cannot guarantee confidentiality**. I must provide our Title IX coordinator with relevant details such as the names of those involved in the incident.

A Comment on Generative AI

We're going to be talking about the role of AI assistants in education for the next decade, and the truth is that no one knows at this moment what we're going to do about it. You are the first generation that this is impacting, and not just in college, but likely when you leave as well.

This can be a good thing. You have the ability to shape your relationship with these tools (even if that means you choose not to use them, or even to fight against their use). But as this is happening, we still need to run courses at universities.

It would be drastic for me to prohibit the use of AI assistants, but they **should not be used to solve homework assignments directly**. I've seen some very powerful and creative ways that they have been used to aid learning. Some ideas:

▷ Ask explicitly for the solution not to be given

▷ Ask about high–level ideas, like analogies or mnemonics

▷ Ask for explanations about how code works, or why a piece of code is not working

If you have any other suggestions please let me know. The point is, it's possible to use these tools in productive ways. Remember, **they're not perfect**, they can be wrong, it can sometimes be easy to tell when a solution was produced by an AI assistant, and you won't have access to them in exams.

Student Resources

There are so many BU resources that it can sometimes be overwhelming. Here's a small list of the ones I think are important to know about, and I'll just add: if you're struggling in this course due to personal/health conditions, I can't guarantee I can help, but if you're comfortable reaching out, feel free to send me an email and I can see if I can point you towards the correct resources. If you're not comfortable reaching out to me, that's okay too, hopefully this list can help you find what you need. Also, keep in mind you can post anonymously on Piazza if you want to ask for help without including your name, and I can do my best to help.

▷ Disability and Access Services

▷ Student Health Services

- Health Promotion and Prevention
- Behavioral Medicine
- SARP

▷ Educational Resources Center

▷ International Students & Scholars Office