98-008: Student Taught Courses (StuCo): Intro to Rust Lang

Updated January 05, 2026

Instructors { Stephen Mao `stmao@andrew.cmu.edu`; Hugo Latendresse `hlatendr@andrew.cmu.edu`; Anish Pallati `apallati@andrew.cmu.edu`

TAs { Max Wen `maxwen@andrew.cmu.edu`

Spring 2026

Office Hours: after lecture, other times TBD	Web: `https://stuco.rs`
Time: Wednesday 7:00-7:50pm	Location: POS-153

Course Description

This course is an introduction to the Rust programming language, a memory-safe systems programming language. We will cover a range of topics, from the Rust borrow checker to parallelism and the `unsafe` keyword. Outside of lectures, there will be programming exercises (homework assignments) that students will need to complete in order to pass this course.

This course is taught in Rust and you are not expected to have prior exposure to Rust. However, this course assumes knowledge of topics covered in **15-122** or equivalent experience with the C programming language. Knowledge of functional programming idioms taught in **15-150**, as well as knowledge of systems programming taught in **15-213** is a bonus, but not strictly required.

Learning Objectives

By the end of the semester, students should be able to read, write, and reason about Rust code. This includes:

- Core Rust syntax such as `struct`, `enum`, `impl` blocks, pattern matching, generics, and `trait`s
- Common Rust types such as `Vec<T>`, `String`, `&str`, `Option<T>`, and `Result<T, E>`
- Use of the borrow checker through ownership and `move` semantics
- Advanced Rust features such as iterators, lifetimes, closures, and smart pointers
- Advanced Rust patterns such as parallelism, concurrency, and `unsafe`

Additional Materials

Access to a computer capable of running `rustc` (the Rust compiler) and `cargo` (the Rust package manager) is required for this course. If you are using stable versions of MacOS, Windows, Nintendo Switch, or Linux, you should be fine.

You can follow the steps from the official Rust website. If you need any assistance, please do not hesitate to ask us!

Course Schedule

Week	Topics *(subject to change)*
Week 1	Introduction
Week 2	Ownership (Part 1)
Week 3	Structs & Enums
Week 4	Collections and Generics
Week 5	Error Handling and Traits
Week 6	Modules and Testing
Week 7	Closures and Iterators
Week 8	The Rust Ecosystem
Week 9	Ownership (Part 2)
Week 10	Lifetimes
Week 11	Smart Pointers & Trait Objects
Week 12	Unsafe
Week 13	Parallelism
Week 14	Concurrency

Attendance Policy

Please read this entire section, as not reading it may cause you to fail this course.

Attendance is mandatory.

According to StuCo guidelines, if you have 3 unexcused absences, you will automatically fail this course.

In other words, if you miss 3 lectures without getting an excuse from us, the StuCo board *requires* us to fail you.

This means that you can miss *up to* 2 lectures at most.

Please let us know if you are going to miss a lecture in advance (at least 2 hours before)! All it takes is a private post on Piazza.

Note that you should **not** email us for an excusal, as emails can be lost / forgotten quite easily. Simply ask us on Piazza.

Participation

We will take attendance at the *end* of each lecture. However, if we notice that many people are arriving more than 10 minutes late, we reserve the right to take attendance at the beginning of the lecture instead.

If you need to miss a lecture, we will determine an appropriate alternative so you don't miss out on that week's content. Most of the time, this is simply reading over the lecture and finishing the weekly programming assignment.

Grading

Grading is determined solely by homework assignments. Most of these assignments are autograded on Gradescope, but there may be a few manually graded assignments that are worth more in the second half of the semester.

In order to pass this course (grade letter P or S), you must accumulate 1000 points. By default, homework assignments are worth 100 points each. However, many of the homework assignments have up to 50 extra credit points. Manually graded assignments can be worth double a normal assignment, but will be graded much more harshly. There will also be many opportunities for extra credit through submission of feedback forms to us.

Grade	Points
NP	0-999
P (or S)	1000+

By reading this syllabus and completing the Gradescope Syllabus Quiz, you will receive 50 points immediately!

Homework

There will be weekly homework assignments where you will write Rust code applying topics we have discussed in lecture. Homework assignments are designed to take on average around an hour per week. **If they take significantly longer than an hour, please let us know!**

All homework assignments are due the week after they are released, right before class starts. Homework assignments will be submitted and auto-graded on Gradescope.

Solutions

We expect you to turn in a submission for every homework, as we believe that the best way to learn a programming language is to *actually write code*. You can only learn so much from watching someone else use a language, and this is especially true for Rust.

However, we understand that you are all busy students and we do not want this course to be a burden. **Therefore, we publish all homework solutions on our course GitHub page.**

There are a few consequences of this decision. The most obvious is that you will have access to a correct solution before you even start. We **strongly encourage** students to avoid looking at these solutions before a good-faith attempt at finishing the homework assignments themselves. You will not learn much by copying and pasting our solutions as your own (and we will be able to detect if you do this).

Referencing these solutions should be a *last resort* for when you either do not have time to finish the homework assignments, or if you are somehow unable to get in contact with us to ask for help. After you complete an assignment, we *encourage* you to look at the reference solution to observe a different way of approaching the problem!

AI

We recommend turning off any AI code assistants or agentic editors for this course, especially in the first few weeks. Learning Rust may require you to rewire parts of your thought process, and that rewiring is hindered when something else does your work for you.

Our assignments are designed to be short, and so they are trivially easy for many of the LLMs you have access to. Remember that the goal here is not to pass the autograder, it is to learn a new programming language!

Late work

Students have 7 late days to be used for any assignment. **However, if you let us know before an assignment's deadline, you can request additional late days.**

Time Commitment

This is a 3-unit StuCo course, which means that this course should require around 3 hours of your time a week, including the 50-minute lecture.

If you find yourself spending more than 3 hours, please let us know immediately so that we can either help you or adjust something in the course itself.

Passing

In the case that a student is unable to reach the passing boundary for any reason, they can reach out to us and we will make sure to work with them and provide alternate paths towards passing (usually a student-defined final project in Rust).

In other words, there are only two ways to fail this course. The first is accumulating three unexcused absences, which is due to StuCo guidelines. The second is the combination of not completing homework assignments *and* not reaching out to us (you will have to reach out to us if you want to do a final project).

Please always make sure to reach out to us on Piazza if you have *any* concerns!

Course Notes

Academic Integrity

Discussion of homework problems, written or verbal, is not only permitted, but encouraged.

That said, **verbatim or near-verbatim copying of answers over any medium is expressly forbidden**. This **includes copying and pasting our solutions**. **If we find that you have plagiarized, we are required to report you to the university.**

If you find yourself needing to do this, talk to us! It might be the case that a change needs to happen on our end, not yours. It is *much* easier for us to give you an extension than to go through the process of academic integrity violations (which we have learned by experience).

Accommodations for students with disabilities

If you have a disability and have an accommodations letter from the Disability Resources office, we encourage you to discuss your accommodations and needs with us as early in the semester as possible. We will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, we encourage you to contact them at `access@andrew.cmu.edu`.

Statement on student wellness

Take care of yourself. Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is almost always helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call `(412) 268-2922` and visit their website at `http://www.cmu.edu/counseling/`. Consider reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

Statement on Diversity, Equity and Inclusion

We must treat every individual with respect. We are diverse in many ways, and this diversity is fundamental to building and maintaining an equitable and inclusive campus community. Diversity can refer to multiple ways that we identify ourselves, including but not limited to race, color, national origin, language, sex, disability, age, sexual orientation, gender identity, religion, creed, ancestry, belief, veteran status, or genetic information. Each of these diverse identities, along with many others not mentioned here, shape the perspectives our students, faculty, and staff bring to our campus. We, at CMU, will work to promote diversity, equity and inclusion not only because diversity fuels excellence and innovation, but because we want to pursue justice. We acknowledge our imperfections while we also fully commit to the work, inside and outside of our classrooms, of building and sustaining a campus community that increasingly embraces these core values.

Each of us is responsible for creating a safer, more inclusive environment.

Unfortunately, incidents of bias or discrimination do occur, whether intentional or unintentional. They contribute to creating an unwelcoming environment for individuals and groups at the university. Therefore, the university encourages anyone who experiences or observes unfair or hostile treatment on the basis of identity to speak out for justice and support, within the moment of the incident or after the incident has passed. Anyone can share these experiences using the following resources:

- Center for Student Diversity and Inclusion: `csdi@andrew.cmu.edu`, `(412) 268-2150`
- Report-It online anonymous reporting platform: `https://reportit.net`
  - username: `tartans`
  - password: `plaid`

All reports will be documented and deliberated to determine if there should be any following actions. Regardless of incident type, the university will use all shared experiences to transform our campus climate to be more equitable and just.