SYLLABUS – CS 536: Foundations of Computer Science   
Instructor: Rose Bohrer (pronouns: she/her)  
Time: T,Th | 4:00 PM – 5:20 PM     
Classroom: Higgins Labs 114  
Instructor Office: Fuller Labs 139

Sections About People (The Important Stuff)

Inclusion + Classroom Climate Expectations

Computer science is for all of us. That means you too. It is important to take a moment to say this because I can guarantee there are students in this class who do not always feel welcome here. Maybe even you.

Inclusion only gets better when people work together to make it that way. I expect all of you to welcome each other, in your own way. On my part, I hope to create a classroom climate where you would feel comfortable sharing any feedback you have on this topic, and also feel comfortable reaching out for help if you do experience any ignorant behavior. Though I have not lived through all the things that my students live through, I know what it is like to look around the room and see that nobody looks like you. I will do my best.

To be clear, here is an incomplete list of things that need to be said more in CS classrooms: All Black Lives (Still) Matter, Anti-Asian Racism is a Pandemic, No Human is Illegal, and Disability Rights are Human Rights.

Mental Health (is in the Back)

I have put my mental health statement at the end of the syllabus to avoid triggering anyone. I encourage you to read it when you are in the right mood because mental health is more important than school.

Disability + Accommodations

The best way to arrange accommodations is to have OAS (Office of Accessibility Services) [notify me](https://clockwork.wpi.edu/ClockWork/custom/misc/home.aspx) – this helps me keep everything organized. It is standard to remind you of this on the syllabus.  
**OAS Email:** AccessibilityServices@wpi.edu **OAS Phone:** (508) 831-4908

***Do you want to know what accommodations are or whether you can get them?*** OAS and your doctor are good people to ask. I’m also happy to give my non-professional advice for things I personally live through: autism, ADHD, Ehlers-Danlos Syndrome, transphobia, and misogyny.

***Is there an accommodation that would help you, but OAS does not usually provide?*** I would love to discuss with you. I can’t promise anything, but we should try.

### Stupid Question Policy

It is course policy that there are no stupid questions. If you still feel that your question is stupid, it is course policy that you should ask it anyway. This policy matters because when you need help, you may not be sure how to ask a perfect question yet. You may also have questions that relate to background knowledge or life outside of class, and these might be even more important. This is a list of questions that are not stupid. It is not complete (there are no stupid questions):

* What happened in the last two weeks of class?
* Is my project idea dumb?
* What do I do if English is hard?
* What does the school counseling center really do? Does it really even help?
* I’m having a hard time in school. What are accommodations? Do I count?
* How do I do well in school if I work two jobs and I’m still hungry?
* How am I supposed to be motivated when **<\*waves hands wildly at the world in general\*>**

I cannot do my job well if my students are afraid to ask for help.

Deciding Whether to Take the Course

What’s This Course About?

The scope of this course is broader than the course description may imply: we cover a broad range of perspectives on programming language. This does not mean that it covers debates like “Should we program in language X vs. Y?” Rather, I cover: what do different kinds of people care about when they think about programming languages? How do researchers in theory, software engineering, social sciences, and humanities ask different kinds of questions about programming languages?

What Will I Do In This Class?

You will do the following in your homework assignments:

1. Implement parts of a programming language (parsing, evaluation, type-checking) in the Rust programming language
2. Perform miniature language design exercises
3. Perform a usability study about language design with your classmates
4. Engage critically with academic research about the intersections of human-computer-interaction with social issues as it pertains to programming languages.

Learning Objectives

Through the above activities, we aim to meet the following list of course objectives:

1. Identify problems where programming language design can be used
2. Communicate with clarity and technical depth about language design
3. Develop a mathematically-precise definition of a language’s syntax
4. Develop a mathematically-precise definition of a language’s semantics
5. Implement interpreters for programming languages
6. Situate your own work among the schools of thought discussed in class

Why Should I Take It?

Potential reasons include:

* You will work as a programmer. Most people don’t write compilers for a living, but most career programmers will use language design skills one day, because most big programs contain “little languages” – features that use these same design skills
* You will work in human-computer interaction or other fields that rely substantially on design skills
* You want to be prepared to perform or consume research about programming languages

How Do Grades Work?

**TODO REVISE**

The grade breakdown is:

- 75% assignments

- 5% attending and participating on the day we perform user studies in class.

- 20% quizzes

- up to 3% bonus credit if you give me feedback on the course textbook **and** **if** I use it

The majority of the assignment work is auto-graded programming work where the score is based on how many tests your code passes. On each assignment, a minority of the work is design work, which is graded by completion, i.e., the percentage of tasks where an honest effort is made.

To provide significant flexibility without compromising rigor, all grades follow the **Honest Effort Policy:**

* If you make an honest effort on an autograded assignment /submission by the deadline, you gain the privilege to resubmit the assignment as many times as you want, at any time up to the last day of class, without grade penalty.
* All homework which is not autograded, is graded based on whether an honest effort is made.
* We wish to reward students who reach out to course staff for help when needed. If you attend office hours or otherwise connect with course staff for help on an assignment, we will always consider that assignment to be an honest effort.
* If a student makes an honest effort on every assignment and quiz but would otherwise have a failing grade, they pass the course with a grade of C. I do not award A’s for effort but I do award C’s for effort.

Extension Policy

Because of the honest effort policy, I expect that requests for extensions will be rare. That being said, I cannot remember the last time that I said no to an extension outright. I will typically say yes to extension requests, but you must request before or as soon as possible to the deadline so that I can serve as a source of accountability for you. That is, if you fall asleep and miss a submission deadline, I will accept the late submission, but if you do not participate consistently in the course and then attempt a late submission of all work at the end of the term, it would not be accepted.

Course Details

Textbook(s) / Learning Materials

I am in the process of developing my lecture notes into a free e-textbook “Human-Centered Programming Languages.” The draft of this book will be your textbook for the course. The textbook includes a significant number of references to other works; if and when you need additional material, you are encouraged to start by following the references. You are also encouraged to tell me about additional material you would like to see in the book, to inform my revisions of it.

Do I Need Any Software?

You will need to install the Rust language and an editor for it. Visual Studio Code is the typical editor.

### Help Hours/Office Hours

**For CS 536: Mondays, 4-5pm, in-person in FL 139**

I also have several other help hours each week for my other students. They get priority at those times, but you are welcome to show up during those times if I’m free: Wednesday 3pm-4pm (in-person in FL 139) and Thursdays, 3pm-3:50pm (on Zoom: <https://wpi.zoom.us/j/4019031992>)

***What are They?*** Office Hours, also called Student Hours, are time specifically set aside for the instructor (me) to give hands-on help to students (you). Office hours are one of the main and first places you should go if you would like help understanding anything in the course, both for help with a specific task or general help. One advantage of office hours is that you can ask for direct or detailed help with course work. Although we spend most of the office hours time talking about things from the course, they can also be a good opportunity for unstructured learning. If you are curious about anything such as related topics or current research in this area, feel free to ask.

***What if I Need More Help?*** I know there will limited scheduled hours of office hours each week, so it can be hard to get enough individual help in a busy week. I want to help you despite these limits, so here is my advice for how to get help despite our limited resources:

* If my office door is open, you can always ask whether I am free to answer a question.
* You are always welcome to ask questions on the course Slack at any time. You are encouraged to ask questions early and often.
* You can post the question on Slack where other students can see it and even answer it. This also reduces the number of duplicate questions and helps me answer you faster
* I will check Slack at least once every weekday during business hours. I usually answer it much more, but questions asked at night are usually not answered until the next day.
* Using Slack helps me be more available to help you
* If you want more scheduled office hours, tell me. I cannot promise them, but I will try.
* If you want an individual meeting for 1-on-1 help, tell me. I may respond by scheduling general office hours instead, but it's good to ask because sometimes I can do 1-on-1.
* If you want a specific kind of office hours such as a review session, tell me.
* If you're not sure how to start, here are [tips](https://ceils.ucla.edu/wp-content/uploads/sites/2/2018/05/Meeting-With-Faculty-In-Person.pdf) for how to ask questions that get good answers.

Schedule

Attendance

Attendance is only explicitly required on days we have quizzes and the day we perform user studies in class, because this is essential for your homework assignment. If you must miss those days, arrange an alternative solution with me by email. On all other days, it is an expectation that you will attend most days and only miss class when you have a reason to do so. However, I will not ask you to give me a reason or to tell me explicitly when you miss class. Just attend by default and miss when you need to.

Your Final Grade

This course is ungraded. This means that you grade yourself based on how well your final project meets the learning objectives for the course. The final project handout on Canvas has the details.

I create an environment for you to learn; you tell me your grade at the end. I occasionally increase students’ grades above their self-grade, if they’re too hard on themselves. I have never reduced a grade below the one a student submits, and never intend to do so.

The only case where I would reduce your grade is if you consistently create a hostile or bigoted environment for other students after I warn you. This is what it means to value inclusion. I have never used this rule and expect it to stay that way

Where Do I Find The Course Content Online?

Canvas. I do not have a separate course website.

Technology Policy

Some days, I will work through examples on my computer in class, so you are welcome to follow along on your computer in class. A computer is not mandatory. You are welcome to goof off on your computer – I too was once guilty of doing homework for one class during a second class to stay awake, and it wasn’t a bad class either. If you do use electronics for primarily entertainment purposes, try to sit somewhere that it won’t distract your classmates, though.

Academic Integrity Policy

WPI’s website has university-wide academic integrity policies. However, by the nature of this course, it is pretty much impossible to cheat. Pretty much the only way to cheat is if your team handed in another team’s project, which I would notice very easily.

# Mental Health

Content warning: transphobia, racism, xenophobia, suicide, and police brutality.

I started this job in Fall 2021. My first year here coincided with the peak of the worst mental health epidemic in WPIs history, with the deaths of 7 students in 6 months. Most of the deaths were by suicide and most of the victims were in computing-related majors.

On a personal level, I want my students to know this profoundly shapes how I view the job. When compared against your personal well-being, academic achievement ***does not matter to me.*** My favorite part of my job is the opportunity to support you as a person, and mental health is key to that. I don’t want to just repeat some boilerplate, so here are some things I want you to know instead:

1. I have been there. There is nothing you can say or do that will scare me. I mean that in a good way.
2. Social connection is good for you, so I try to encourage social interaction in the course design.
3. I’m not going to judge you, period, I just want you to be okay. Missed class? Don’t know what’s going on? Cried in lecture? Vomited on my office floor? I’m not going to judge you. ***I need you to internalize this fact.***

I have been encouraged to tell you to contact the counseling center for non-emergency treatment ([sdcc@wpi.edu](mailto:sdcc@wpi.edu), [508-831-5540](tel:508-831-5540), 16 Einhorn Road) or call Campus Police ([508-831-5555](tel:508-831-5433)) if someone’s life is in immediate danger. However, nobody is stopping me from also telling you:

1. There are many therapists in Worcester that take student insurance
2. General practitioners (“your regular doctor”) are the main source of antidepressant prescriptions.
3. Calling American police during mental health crises has gotten innocent patients killed, especially Black, Brown, trans, and disabled patients. They have traumatized people who I personally cherish. Mobile crisis response teams are an alternative and are often much better at avoiding violence. Community HealthLink has one in Worcester. I do not know whether it is good, but I feel I should share a link: <http://www.communityhealthlink.org/chl/youth-and-family-services/youth-mobile-crisis-intervention-ymci>
4. The Trans Lifeline (877-565-8860) is a crisis-intervention hotline (“suicide hotline”) run by binary and nonbinary trans people. As of 2020, they do not call police unless required by law.

The family of one of the victims (Nathan S. Morin) called for donations to [American Foundation for Suicide Prevention](https://supporting.afsp.org). In his memory, consider this a standing invitation to support mental health orgs in our community.

In conclusion: I think about mental health a lot. Like, a lot a lot. Don’t be shy, okay?

What would useful industry stuff look like?  
- Config files  
- Code autogeneration  
- Mini languages for specifying permissions / sets  
- CSV

What would assignment timeline look like?

1. Project Planning
2. Syntax design + Parser-generation
3. ASTs + Pretty-printers + basic checks
4. Goals and evaluation and study results
5. What’s it mean? What semantics should t be anyway?
6. What should its types be if any? Who cares?
7. Reflect
8. Queer it
9. Have fun
10. Done

Week 1: Have Fun, Pick Project Teams

* Introduction
* Topics + Teams

Week 2: Learn How to Plan and Evaluate PLs

* Evaluation of Languages
* Kinds of languages + impl approaches

Week 3: Syntax Review: RE + CFG

* RE
* CFG

Week 4-5: Learn (Functional?) Programming? In Scala?

* Fastparse
* Precedence climbing
* Datatypes
* HOFs
* Pretty-printing generators

Week 6: How to User Study – take from CMU class

Week 7: Study + Reflect + Plan

Week 8: Semantics

* Operational basics
* Operational hard version

Week 9: Types

* Boring type system
* Parametricity idk lol

Week 10: Dialectics

* Write my paper by now
* Explore what properties of your lang can/cannot be done
* At least a week of queer theory
* Feminist programming languages
* Why do trans women write rust
* Redo entirely

Week 11: Beauty

* Processing
* Penrose
* FARM papers
* Failure continuations as queer play

Week 12: Play

* Inform
* Twine

Week 13: Distributed

* Parallel semantics
* MLton Stefan stuff

Week 14: TBA+Present

My ideas:  
1. Idea picking: Pop charades

2. Functional programming, why

3. Evaluation framework for languages

4. Rhetorical evaluation of languages

5. User studies for languages

Distributed?

This course discusses the fundamental concepts and general principles underlying current programming languages and models. Topics include control and data abstractions, language processing and binding, indeterminacy and delayed evaluation, and languages and models for parallel and distributed processing. A variety of computational paradigms are discussed: functional programming, logic programming, object-oriented programming and data flow programming. (Prerequisites: student is expected to know a recursive programming language and to have an undergraduate course in data structures.)

<https://www.cs.cmu.edu/~aldrich/courses/17-396/>

|  |  |
| --- | --- |
| Jan 14 | Course introduction |
| Jan 16 | [Concepts for Language Design](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/lec2-concepts.pdf) |
| Jan 21 | Project proposals |
| Jan 23,28 | Formalism, e.g. Featherweight Java ([paper](https://www.cis.upenn.edu/~bcpierce/papers/fj-toplas.pdf)), [Evaluation and Typing Derivations (.txt)](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/evaluation-and-typing-derivations.txt) (view in fixed-width font) |
| Jan 30 | Soundness; Assembly-like IR (see sections 2.2 and 3.1.3 of my [program analysis book](https://www.cs.cmu.edu/~aldrich/courses/17-396/program-analysis.pdf)) |
| February 4 | [JavaScript](https://www.cs.cmu.edu/~aldrich/courses/17-396/js/); calculator example ([calc.zip](https://www.cs.cmu.edu/~aldrich/courses/17-396/assignments/calc.zip), [web page](https://www.cs.cmu.edu/~aldrich/courses/17-396/assignments/calc/)) |
| February 6 | [Implementing an Interpreter in JavaScript](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/lecture8-notes.txt) (coverage of Assignment 6 constructs was deferred to Feb 11) |
| February 11 | [Big-step environment semantics](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/big-step-env.pdf) |
| February 13 | [Introduction to User-Centered Programming Language Design](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/HCPLD1.pdf) |
| February 18 | [User-Centered Programming Language Design, part 2](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/HCPLD2.pdf) |
| February 20 | [User-Centered Programming Language Design, part 3](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/HCPLD3.pdf) |
| February 25 | [User-Centered Programming Language Design, part 4](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/HCPLD4.pdf) |
| February 27 | Internal Domain-Specific Languages |
| March 3 | Case Study: The Design of the Obsidian Language |
| March 5 | Parsing |
| March 19 | [Pure OO PLs and Transpilation](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/lec17-oo.pdf) |
| March 24 | Transpilation Demo ( [txt notes](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/transpilation-notes.txt), [manual c translation](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/prototype.c), [generated c](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/gen.c), [transpiler impl](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/transpiler.zip)) |
| March 26 | [Corpus Studies for Programming Languages](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/corpus-studies.pdf) |
| March 31 | [Case Studies on Programming Languages](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/case-studies.pdf) |
| April 2 | [Language Expressiveness](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/expressiveness.pdf) |
| April 7 | [Truffle: Partially Evaluating Interpreters](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/truffle.pdf) |
| April 9 | [Glacier: Usable Enforcement of Transitive Immutability](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/glacier.pdf) |
| April 14 | [Penrose: From Mathematical Notation to Beautiful Diagrams](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/penrose.pdf) |
| April 16 | Discussion: User Study results |
| April 21 | [Wyvern: Designing a Language for Security](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/wyvern.pdf) |
| April 23 | [The History and Impact of OO: Simula and Smalltalk](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/oo-history.pdf) |
| April 28 | [Programming Language Design and Performance](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/language-performance.pdf). See also [this lecture](http://www.cs.cmu.edu/~aldrich/courses/17-355-18sp/notes/slides20-declarative.pdf) in my program analysis course for an example of getting high performance from using a DSL. |
| April 30 | [Perspectives on PL Design and Course Conclusion](https://www.cs.cmu.edu/~aldrich/courses/17-396/slides/perspectives-conclusion.pdf). |
| May 7 | Final project presentations |

Fisler (20 YEARS AGO):  
  
2x: overview

2x: first-order, data rep

2x: impl first-order

2x: closures, mutation, recursion

2x: recursion, exceptions, contexts, letcc

2x continuations

2x: semantics, lambda calculus

2x compilation

4x garbarge collection

2x types

2x type inference

2x polymorphism

2x scripting

https://cseweb.ucsd.edu/~mcoblenz/teaching/291I\_fall2022/