SA/TA/GLA Instructions – CS 4536: Foundations of Computer Science   
Instructor: Rose Bohrer (pronouns: she/her)  
Instructor Office: Fuller Labs 139

# Overview

This document defines the job expectations for course staff, to help support communication and reduce stress for all of us.

This class is a unique style of programming course. Half of it is like a traditional 4000-level programming languages elective, half of it is human-computer interaction material focused on the design of programming languages and their social implications. The traditional material includes topics like mathematically defining the semantics of a programming language, designing static type systems formally, and designing and implementing technologies like parsers, type-checkers, and interpreters. Course staff are not expected to be equally strong in both aspects of the course as long as they are strong in one of the two areas.

There is one final exam and no midterm. There are 5 homework assignments. Every homework assignment has two parts: a programming assignment in the Rust programming language and a written assignment. Together, the five programming assignments implement a small programming language. Together, the five written assignments have each student design a small user study about programming languages which they run on their classmates during class time and then interpret the results of.

The class is designed for scalable grading. Programming assignments are autograded and written assignments are graded by completion (full points if you make a real attempt). For this reason, your grading time will focus more on giving open-ended feedback than on rubric grading.

**Terminology note:** There are many different job titles for course staff and it is important to use precise terminology because certain job titles come with certain protections under union contract:

* Student Assistant (SA) is typically an undergraduate student hired by the department for course support at up to 10 hours per week across different courses in different terms
* An undergraduate grader (UG) is typically hired for a specific course for a specific term at up to 10 hours per week
* A Teaching Assistant (TA) is a graduate student hired by the department on an ongoing basis for up to 20 hours of course support per week.
* A Graduate Learning Assistant (GLA) is another kind of graduate student hired for up to 20 hours of course support per week. Unlike TA’s, GLA’s do not receive tuition remission from WPI, and on the flip side there are certain restrictions on which tasks can be assigned to GLA compared to TAs.

If there is ever any uncertainty about your job title, please contact myself, the TA/SA coordinator, or the graduate coordinator so we can clarify.

# Time Breakdown

Graduate course staff are expected to spend an average of 20hrs per week supporting the course. Undergraduate course staff are expected to spend an average of 10hrs per week. Here is an approximate schedule for undergrad staff:

**Note:** The undergrad schedule has fewer hours for learning material because it is written for those who have taken the course previously from the same instructor.

* 1hr – Attend staff meeting
* 1hr – Respond to student questions / emails online
* 2hr – Review course material
* 2hrs – Hold office hours
* 4hrs – Provide feedback on student work (grading homeworks / exam)
* Once a term: help proctoring final

Here is an approximate schedule for graduate staff:

* 1hr – Attend staff meeting
* 1hr – Respond to student questions / emails online
* 3hrs – Hold office hours
* 5hrs – Learn the course material
* 10hrs – Provide feedback on student work (grading homeworks / exam)
* Once a term: help proctoring final

# Staff Meetings

We will meet weekly to discuss the state of the course and plans for the next week. Meetings will not always necessarily take a full hour. During these meetings, you should also reflect on your interactions with students and help share their feedback, feelings, and thoughts with me.

# Providing Feedback on Student Work

There are 56 students this year, meaning you can spend about 15 minutes on each submission without going over your working hours limit. For programming assignments, focus on:

* Fundamentals of good code style. There is no need to picky, the goal is to write code that is easy for you as the grader to read.
* Did they make the task harder than necessary? For example, did they reimplement a feature provided by the libraries?
* If they did something nice beyond what we ask for, leave a kind comment. For example, some students might use higher-order functions to make code cleaner when it is not required.
* If they are way off (for example, the assignment asks to write a recursive function over syntax trees, but they did not use recursion), try to provide a specific suggestion about what they should study or review (specific sections of the book or slides)

For written assignments (which include students leaving comments on each other’s work), focus on:

* Are they on track to be able to run a small user study on their classmates toward the end of the course?
  + Lectures on study design start halfway through the term, so their early work can be high-level and this is okay
  + At the same time, as course staff, you can help ask questions that will lead them toward working out the details of their user study
* Are they taking the peer review / comment task seriously? In any class with required discussion, it is common for students to make boring comments just to get the credit. If you see this happening, leave a comment encouraging them to provide more feedback, so they know we do care. They are expected to ask at least one specific question to each other or say at least one specific thing to focus on.
* Encourage them to have fun with the written assignment, show their own creativity, and embrace their own interests. This is why we use completion grading for written assignments.

If there are major red flags for cheating, privately contact me about the red flags. Continue grading the homework normally. Don’t spend too much time worrying about cheating and don’t make accusations toward a student since that’s the instructor’s job. Here are examples of potential red flags, though context matters a lot:

* Submitting code that is in the wrong language or relies heavily on strange libraries or syntax that are not discussed in class. This sometimes happens with ChatGPT-generated homework submissions as well as answers looked up online. For example, one of our assignments is to write a parser. If they use a completely different parser library without permission, it’s suspicious.
* Submitting solutions to a previous year’s homework if the assignment changed a lot. I only plan on making significant changes to one or two assignments, though.
* Two students submitting code that makes the same very-rare mistake that appears in no other submissions.
* Some instructors insert invisible text in assignment handouts to poison ChatGPT prompts. I have not decided whether I will do this, but if I do, I will let you know what outputs to look for.
* Two students submitting written assignments that are suspiciously similar to each other, but they have not talked to course staff to let us know that they are openly working on related projects and talking to each other.
* For written submissions, ChatGPT is known to write in a specific authoritative tone using needlessly complex language. It’s hard to be sure, but you can flag this for me to check.

The following are not red flags:

* Submitting code that does not compile due to a compilation error that is common among students (like struggling with the type system)
* Two students structuring their code in a similar way to each other or our staff solution. For most of the programming assignments in this course, there are only one or two good approaches, and correct student submissions will be similar.
* Two students making the same mistake, but it’s a common mistake.
* Seeing students working together, as long as they are not taking each other’s solutions.

# Learning the Course Material

I use my own materials to teach the course, so you should spend time getting familiar with them. If you took the course before, you can review your past work. If not, you should do every homework before your office hours so you are prepared to help students. Use the lecture notes and textbooks to learn-as-you-go to help complete the homeworks.

You are not required to attend lecture. If you would find attending helpful, let me know, but it is possible you would not be allowed to due to restrictions on the number of people in the room.

# Optional: Connecting Your TAing + Research

Sometimes I write research papers related to my teaching work. You are never required to contribute to that research work as part of your TA job. If you wish to be a part of the research project for your own reasons (such as wanting an opportunity for paper coauthorship) you are welcome to discuss with me.

# Time Management + Life Management

Course staff are expected to communicate openly and honestly if you are struggling to find time to complete your assigned work. I understand that you are students with many different responsibilities, which you cannot always predict. Sometimes you will have an important deadline outside the course, or you will get sick, or you will need to travel, etc. My job as the instructor is to be able to adjust when somebody runs into trouble. Open communication is important because it allows the rest of us to support you when needed.