Assignment 2 – CS 4536/536: Programming Language Design

Instructor: Rose Bohrer (pronouns: she/her)
Instructor Office: Fuller Labs 139

Overview

The primary goal of the programming assignment is to develop a parser for a simple programming language, i.e., a program that recognizes the syntax of a simple language.

How Long Will This Take?

My goal is that most students will spend at most 13 hours/week on this course and that few will spend more than 15 hours/week, in total.

If you have spent more than eight hours on this assignment and are not done, you are **expected** to reach out to the course staff for help (office hours, chat, or email). It is our job to help you and your job to seek help when appropriate.

Grading

This assignment is worth 100 points. 75 points are for the programming part and 25 points for the written part. Your grade for the programming part is the sum of scores for each passing test case. Your grade for the written part is 20 * the fraction of problems for which you submit an honest attempt at a solution plus 5 * the fraction of assigned peer reviews you complete.

What to Read While Working on This Assignment

In addition to the lecture readings about Rust, you should refer to the official documentation for the Rust peg crate. You are welcome to refer to other resources about parsing expression grammars.

Programming Assignment

Download and extract the programming assignment from Canvas. Your assignment is to implement all functions (and parser rules) to meet their specifications as given in the comments. Point breakdown:

- 10pts – identifiers (e.g. variable names)
- 5pts – numerals (i.e., numeric literals)
- 21pts – other expressions, without definitions
- 39pts – definitions and expressions that contain definitions

The handout lists the length of our staff solutions to help you decide if your approach is more complicated than expected. For this assignment, the number of lines is small, but we expect it will take longer than the line count suggests, as it relies on a new library with a custom, very concise syntax.

Written Assignment

Over the next few written assignments, you will prepare (and then perform, in-class!) an experiment (user study) to answer a question that you personally have about programming languages.

Note: You will peer-comment on each other's answers, as part of the following assignment. Do not submit anything that you would want to keep private from classmates.

Though you will receive points for completing peer feedback, the feedback does not influence the classmate's grade, which is assigned based on completion. The goal is to focus on writing your proposal in a way that would be fun for you or a classmate to read.

20pt Submit a pitch for an experiment. This typed pitch should contain all of the following:

- A high-level question that you are interested in answering about programming languages (in general, or a particular language)
- Describe why you are interested in the problem
- A break-down of the high-level question using the School-of-Thought archetypes. List all archetypes that can help answer the question, and identify which parts they would help with
- You will act as a Social Scientist to answer the question in a future assignment, so make sure the Social Scientist can contribute significantly to the answer. Make sure to include at least one falsifiable, testable hypothesis that could be confirmed or denied by experimental user tests
- Describe how you would imagine testing that hypothesis, especially in a classroom setting. While we may announce a different testing format later, we expect to spend about an hour of class time on conducting studies, with about 10 minutes for each student to conduct their studies on the other students who share their table. To make the best use of time, you're strongly encouraged to design your study so that all the neighbors at your table can participate at the same time as each other.
- If (and only if) conducting your study will require you to use new material outside what we study in class, either (1) provide a plan on how you will learn that material or (2) if you already know that material, write about in a way that will help your reader both understand the material and understand that you understand it.
- There is not a specific length requirement, but we expect most responses to be about 0.5-1 page. What is important is that you satisfactorily answer all the questions. Remember that you will actually carry out a study based on your question in class with your fellow students, and that giving serious thought now will save you effort in the long run. Your pitch will be peer-reviewed in the next assignment and you should write for your classmates as your audience.

Read before doing this task: The following task is to practice giving feedback on an example proposal written by the instructor. You are welcome to read that proposal to get used to the format.

Language Design vs. Study Design: Many students may be interested in designing a new language or language feature. You are only required to design a study, but that study could use either an existing language or a new one. If you design a new PL in your study, talk to course staff for advice on scope.

Open-ended: If you're interested in doing something a bit different, you are allowed to deviate from this format as long as you provide a solid plan and will exercise similar core skills. If you deviate significantly, do chat with course staff to help provide a solid plan.

5 pts Practicing for Peer Reviews

In future assignments, you will give rubric-based constructive feedback to each other on your proposals. "Incomplete" The work is not complete enough to provide quality feedback.

In your write-up, provide written feedback which follows the given rubric:

- Creativity+Motivation: Highlight any particularly exciting or creative ideas in the proposal. If you are not sure what to write, you can look at the flip side: is there any aspect of the proposal where you wish they explained their motivations more clearly?
- Detail: A write-up should be detailed enough that you can provide good feedback. How well does it meet this goal? Do you struggle to understand the core concept? Do you understand the core concept but have uncertainty about the goals? Uncertainty about the approach that will be taken in the study?
- Organization: To provide understanding of the above points, a good proposal not just enough detail but organizes it well. Does the proposal make it easy to identify all the required proposal sections? Do the ideas in the proposal connect together nicely? Is it concise?
- Preparation: This does not apply to all studies. If the study requires any significant preparation outside of the planned course activities, how well do they address this preparation? Did they acknowledge that preparation is needed? Did they cite specific resources they will use? If they are already familiar, did they demonstrate their knowledge?

You should aim to include at least one (short) comment for each bullet point of the list (4 total). You are allowed to give comments outside the rubric, too. Write your feedback using the following guidelines for a professional tone, which are based on National Science Foundation standards:

- Criticism is welcome, but it should not be personal, it should be focused on the work ("the response to question 1 is missing" vs. "you forgot to answer question 1")
- Use "I" language when appropriate ("I didn't understand X" vs. "X makes no sense")
- Avoid superlative words or exaggerating language "very confusing"
- Be specific ("a plan for interpreting user data was not provided," "the proposal is vague")
- Do not imply a person should give up. Reviews should be consistent with growth mindset.

The example proposal starts on the next page. Give feedback on this example using the rubric above.

The example proposal is intentionally not perfect, so that you can practice both positive and critical feedback. It is meant to be a good but not perfect proposal.

Example Proposal: Designing Whitespace

High-level question: I want to understand: "What is the right way to handle whitespace in a PL?" Some programming languages are whitespace-insensitive, others (like Python) use indentation in a significant way, and I'm sure there are other approaches as well.

Interest: My friends and I have different favorite PLs, and we've debated which ones have better whitespace before. It would be a lot of fun if I could finally answer our debate.

Schools of Thought:

- **Social Scientist:** I'm going to have my classmates try out different whitespace and see what happens, using social science skills
- **Theorist:** Since the theorist can include the theory of syntax (regular expressions + grammars), I do think there's a role for theory here. I want to understand if there's a fundamental theoretical difference between the whitespace-sensitive and whitespace-insensitive approaches, maybe whether one of them is more powerful than the other or if they're actually just basically the same as each other.
- **Humanities:** If I have some extra time, I'd actually love to learn from visual design and typesetting, which can be considered part of humanities. I've always thought that traditional print layout and illustration are interesting, and I want to see if this can inform PL design

Testing: I'm going to give each of my classmates 2 or 3 different programming languages with different rules for how whitespace works. I'm going to give them a programming task and have them right down answers to it on paper. I'll have one-on-one interviews to expose any challenges they noticed, then I'll check the correctness of the answers. I'll use both of these kinds of information to evaluate the different kinds of rules for whitespace

New material: I think I can get the basic version of this study working with just the material from the class. I'm a bit nervous about conducting the study because I haven't done that before, so I might read ahead to chapters 10-12 a bit while I'm doing the homework.

I feel comfortable with the theory of syntax we are covering in class, but we didn't talk about whitespace a whole lot in those lectures, so I want to see if there's anything more. I'll look on Wikipedia and the official documentation for different PLs first.

If I get around to the humanities part of the project, that's actually where I need to learn the most new things. I think print layout and illustration are very pretty but I don't have much practice with them myself. I'm not sure who to ask.