CS 330 Homework

Terp

1 Overview

Your task in this homework is to write an interpreter for a domain-specific language of your own design. This homework is very open-ended. You may target mostly any domain. Possibilities include but are not limited to languages that allow programmers to produce images, sound, text adventure games, scientific calculations, shell-like interactions with processes and the file system, webpages, web services, and so on. You are encouraged to make a language that you would actually use.

2 Requirements

Your solution is to meet the following specification:

- Place all files in directory `<YOUR-REPOSITORY>/terp`.
- All code must run on a conventional Linux machine.
- Provide in file `README.md` a description of your language, its grammar (following the model discussed in class), and instructions for how to build and run your interpreter.
- Your language must support variables, conditional logic, and some means of parameterized procedural abstraction (functions). Flow control may be appropriate for your language, but it is not required.
- You must write a lexer and a recursive-descent parser by hand. You are certainly encouraged to reference the code we wrote in class, but we omitted a lot of error-checking and wrote code that needs considerable refactoring. Do not use it without doing significant cleanup. Your syntax should not be identical.
- Your interpreter must not look like it was hastily assembled with the minimum effort required to receive credit.
- Share on Piazza under folder `terp_share` some examples of your interpreter in action. Videos and screenshots are welcomed.

You may target mostly any other language. There is no SpecChecker. You may work with one partner. Include both your names in `README.md`.

3 Later Week

To qualify for later-week submission, a description of your intent for your language and its proposed grammar must be provided in `README.md`.