Lab 7

Due Monday, April 15 by 11:59pm

Handout 17
CSCI 334: Spring 2024

Coding Guidelines

The usual guidelines apply for both programming and LaTeX submissions: be sure to provide all of your source code and make sure that it compiles (20 points).

Turn-In Instructions

Turn in your work using the `git` repository assigned to you. The name of the `git` repository will have the form `https://aslan.barowy.net/cs334-s24/cs334-lab07-<USERNAME>.git`. For example, if your CS username is `abc1`, the repository would be `https://aslan.barowy.net/cs334-s24/cs334-lab07-abc1.git`.

You should have received an invite to commit to the repository via email. If you did not receive an email, please contact me right away!

Group Programming Assignment

This is a partner lab. You may work with another classmate if you wish, and you may co-develop solutions. Remember: although you can work on code together, you must each independently write up and submit your solution. No code copying is allowed. Tell me who your partner is by committing a `collaborators.txt` file to your repository. **Be sure to commit this file whether you worked with a partner or not.** If you worked by yourself, `collaborators.txt` should contain something like "I worked by myself." (5 points)

This assignment is due on Monday, April 15 by 11:59pm.

Reading

1. **(Required)** "Parsing"
2. **(As Needed)** Previous student presentations

Problems

Q1. (25 points) ..... Parsing Basics

Consider the following grammar,

```
<expr> ::= a<B> | ϵ
   <B> ::= b<expr>
```

where ϵ stands for "nothing."

A fellow programmer came up with the following parser, but it's not working the way they want. They come to you for help.

```
let parse input =
    // parser
    let grammar = pmany0 (pstr "ab")

    // parse input
    match grammar (prepare input) with
    | Success(res,_) -> printfn "%A" res
    | Failure _ -> printfn "Invalid expression"
```

The parser correctly accepts everything the grammar says it should, like the strings `"ab"`, `"abab"`, `""`, and `"ababab"`. However, it also incorrectly accepts the following strings: `"aba"`, `"abfoobar"`, and `"Wait, how the heck is this possible?"`. Fix the above program so that it only accepts valid strings from the grammar and rejects all other strings.

The project directory for this question should be called "`q1`". You should be able to run your program on the command line by typing, for example, "`dotnet run`".

Q2. (5 points) ..... Project Partner

If you plan to collaborate with a classmate on your final project, please let me know who that person is by filling out the following project partner sign-up sheet (`https://forms.gle/KKZWJuCugKTyMkrb8`).

Q3. (50 points) ..... Project Brainstorming

For your final project, you will design and implement a programming language. For this part of the assignment, you should think of three different possible final projects you might explore. One of those proposals should involve an artwork at WCMA's Object Lab. The remaining two can be any project that you want. Feel free to let your imagination run wild.

If you find the notion of creating your own programming language daunting, don't worry. For future project checkpoints, you are welcome to fall back to a "default" Object Lab project whose scope will be a little less open-ended. For now, throw caution to the wind with the assurance that at this stage any idea you dream up is non-binding.

Note that a programming language need not be a so-called general purpose programming language. A general purpose programming language is equivalent in expressive power to the lambda calculus or a Turing machine, meaning that is can be used to compute anything. Instead, I encourage you to explore domain specific programming languages, or DSLs. DSLs are usually tailored toward solving a specific problem.

You have probably used a DSL without even realizing it. Some example DSLs are:

(a) `make`

(b) GraphViz

(c) Hypertext Markup Language (HTML)

(d) Extensible Markup Language (XML)

(e) Structured Query Language (SQL)

(f) Markdown

(g) LaTeX

(h) Scalable Vector Graphics (SVG)

(i) JavaScript Object Notation (JSON)

If you have a personal itch, scratch it. For example, I often have to grade student work, and the part I always mess up involves computing scores. Therefore, the grading rubrics I supply to my teaching assistants are actually written in a domain specific language I designed called `tabulator`.

Be creative! Do you love music, or art, or literature? Can you make a language that generates music? Could you make a language that creates art? Could you make a language that generates poetry? Some of your former classmates have designed languages that have done precisely these things and more.

For each potential language, describe

(a) The purpose of the language. What problem does it solve? When I have trouble focusing on this part, it often helps me to think in terms of *inputs* and *outputs*.

   i. *Programs*, written in the language you design, are the inputs.

   ii. The output is what happens *when one of those programs is evaluated*.

(b) Write two sample programs to demonstrate what the language might look like. Feel free not to be constrained by reality at this point. Just imagine that your programming language already exists.

For the Object Lab piece, you can propose anything computational involving that artwork. If you are struggling with this idea, as a "default project," consider the following purpose: your language should generate artwork in the style of the artist. One way to do this would be to either make your program non-deterministic (it "interprets" the program differently every time), or perhaps it takes an input that tells the interpreter how to proceed with variations deterministically (e.g., by specifying a random seed).

For example, for the Josef Albers painting, "Homage to the Square: Warming," one kind of "fantasy program" might be:

```
5 squares, from gray to orange, stacked and 25% smaller, variation 1.
```

With small tweaks to the same syntax, we can also get very different outputs.

```
8 squares, from gray to blue, side-by-side and equal size, variation 2.
```

Observe that for this language is it easy to ask the following questions: What's the grammar? Does it take any additional inputs? What are the outputs? Think about those questions; we will answer them in a future lab. If the answers come easy, it means it's probably an interesting and well-defined language. In the future you will say precisely how your language is parsed and interpreted, so although you don't have to write anything down now, you might spend a few minutes thinking about it.

Be sure to submit your project brainstorm in a file called "`lab-7.tex`" along with a pre-built "`lab-7.pdf`" in a folder called "`project`".

Q4. ($\frac{1}{10}^{\text{th}}$ bonus point) ..... Optional: Feedback

I always appreciate hearing back about how easy or difficult an assignment is.

For $\frac{1}{10}^{\text{th}}$ of a bonus to your final grade, please fill out the following Google Form.