Lecture 1:

- Brief Course Overview
- Intro to MyPL

Homework:

- HW-0 out (setup)

Course Overview

Deep dive into programming language (PL) design & implementation

- implement a "made up" typed, procedural programming language (`MyPL`)
- explore functional programming (using `Haskell`)

General course goals

- More programming experience (using ideas/techniques you've already learned)
- Better understanding of how compilers/interpreters work
- Better understanding of language design (syntax, types, constructs, trade-offs)
- Exposure to different programming "paradigms" (procedural vs functional)

Why study language implementation ("compilers") ...

1. Essential part of computer science (and most computer-science curriculum)
2. Complicated engineering problems (example of how to build larger systems)
3. Techniques useful for a wide range of software development problems
4. Better understanding of how languages work (can improve your programming)

Why study functional programming?

1. Functional constructs have gained popularity in most (non-FP) languages
2. New ways to think about programming, new tools for problem solving

Logistics

1. Course webpage: `www.cs.gonzaga.edu/bowers/courses/cpsc326`

2. Piazza: for Q & A, announcements (see invite)

3. GitHub: for homework assignments

4. Canvas: for tracking grades (but not "official" gradebook)

5. Office hours: MW 3-4:30, F 12-1

6. Grading: 700 points total (see syllabus) ... *note*: **individual work!**

   - Homework: 300 pts ... HW 0 (10), HW 1–6 (40 ea), HW 8-9 (25 ea)
   - Final Project: 80 pts ... MyPL extension
   - Quizzes: 80 pts ... 8 quizzes (10 ea), 1 make-up quiz
   - Exams: 200 pts ... 2 midsemester (50 ea), 1 final (100)
   - Attendance: 40 pts ... $\approx$ 40 lectures (1 ea)

   To pass must score 60% on homework + project **and** *60% on exams + quizzes*

7. HW Late Policy: 25% penalty up to 2 weeks after due date (excludes HW 7, 8)

8. Need: `python` 3.10.12+, `pytest` 7.4.4+, `ghci` 8.8.4+, `git` (GitHub account)

Warning: Programming heavy, HWs build on each other, don't get behind!

Brief Intro to MyPL *(v7)*

Basics:

- Simple PL for exploring design and implementation ideas
- Strongly typed language
- Includes typical base types, functions, structs, and arrays
- Struct and list "objects" are allocated on the heap
- Functions use pass-by-value, objects passed as "references" (object ids)

1. Comments:

```
// this is a single line comment
// only single-line comments are supported
```

2. Primitive Data Types:

```
int       // integer (any size, like in Python)
double    // floating-point (double precision)
bool      // either true or false (not 0, 1)
string    // sequence of characters
void      // used as function return types
```

3. Values:

```
0, 1, 7, 10, 20, 876132     // int values
1.0, 1.01, 10.3, 0.505      // double values
true, false                 // bool values
"foo", "bar", ""            // string values
null                        // legal value (any type)
```

... to be continued