Lecture 1:

- Course Overview & Organization
- MyPL part 1 (basics)

Course Overview

Deep dive into programming language (PL) design & implementation

- along the way, implement a "made up" typed, procedural language
- including parsing, type checking, and execution

General Course Goals:

- more programming experience
- understanding of how PLs (compilers/interpreters) work
- understanding of language design (syntax, types, constructs, trade-offs)
- (brief) exposure to different programming "paradigms"

Why ...

"What I cannot create, I do not understand" – Richard Feynman

Understanding how PLs work ...

- can help make you a better programmer
- can help decrease time needed to learn new languages
- techniques useful for wide range of software dev problems
- example of a more complicated/larger engineering problem
- essential part of computer science (curriculum, PLs as a subfield)

Studying PL concepts generally ...

- new ways to think about programming and problem solving tools
- better understand language and performance trade-offs

Logistics

1. Course Webpage: for course schedule, links (e.g., hw, lecture notes)
2. Canvas: for grades only!
2. Piazza: for Q&A, announcements (see invite)
3. GitHub: for homework
4. Office Hours: MWF 2:10-3:30

New: study "guides" (things to focus on) as part of most lectures

Good Practice: study material after each lecture and before next lecture

Workload

Course work: 600 points

• 8 assignments (points depend on assignment)	... 240 points
• Individual final project	... 60 points
• 6 quizzes (10 ea.), 3 make-ups	... 60 points
• 2 midterms (60 ea.), 1 final (120)	... 240 points

To pass: must pass HW (incl. project) **and** tests (quizzes, exams) separately

Late policy: 30% penalty $\leq$ 1 week, 60% penalty $>$ 1 week late

Quiz policy: no make-ups, last 3 can be used to replace those missed

Important: carefully read the syllabus for additional information

Expectations

Basics:

- participate and be engaged ... no electronic devices in class $^{(*)}$
- start assignments early
- assume you have everything you need (ask when in doubt)
- do your own work

Hints and Warnings:

- set aside time: HWs require signficant time investment (programming)
- come to class, pay attention, participate, take good notes
- go over and study after each class meeting (see "guides")

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

Course Project

1. Due at end of semester (before finals week) ... worth 2 HWs
2. Separate check-ins (see course schedule)
3. Extend MyPL with new constructs / features ... see instructions for details
4. Can do one of two different options for extra credit ... worth 1 HW
   (a) Take a deep-dive into PL research (read 2 papers, present results)
   (b) Learn a new language, write example program(s), report findings

Intro to MyPL

MyPL Basics: ... *a simple PL for learning about PL implementation*

(1) programs as a single-file set of functions (required `main`)

(2) variables and simple arithmetic ops

(3) if, while, and (simple) for statements

(4) structs and 1-D arrays ... as "objects" allocated on the heap

(5) functions use pass-by-value, objects passed as "references" (via object ids)

(6) small set of built-in functions ... I/O, type cohersion, arrays

(7) explict type declarations ... but some type inference for variables

(8) static typing ... type errors detected "statically" (before runtime)

Note: we are using Java 21+ (with Maven) to implement MyPL this semester