Programming Language Concepts

CSCI-344 Section 01
Prof. Arthur Azevedo de Amorim

MoWeFr 2:00pm – 2:50pm
SLA-2150

Programming Language Concepts

Introduction and Overview

Course Description

This course is a study of the syntax and semantics of a diverse set of high-level programming languages. The languages chosen are compared and contrasted in order to demonstrate general principles of programming language design and implementation. The course emphasizes the concepts underpinning modern languages rather than the mastery of particular language details. Programming projects will be required.

Course Goals

Widen perspective and understanding of PLs.

- Alternative programming paradigms and languages
- Theory that underlies language mechanisms
- Implementation choices for implementing languages
- Become a much better programmer
  - (in any language, even languages we won't study)

Course Motivation

Why not study programming languages?

- "I only need to know C++ (or C# or Java or Python or ...) to get a job."
- "I already know how to program."
- "I will never need to design (or implement) a programming language."

Course Motivation

Why not study programming languages?

- "I only need to know C++ (or C# or Java or Python or ...) to get a job."
- "I already know how to program."
- "I will never need to design (or implement) a programming language."

- How many have been on co-op? and used a programming language that had not been introduced in a course?
- Are you ever done learning how to paint? dance? build?
- How many have used a non-trivial library API (e.g., OpenGL)?

Course Motivation

Why study programming languages?

Languages influence way of thinking:

- Learn to learn languages
  - Use more than one language in your career
- Learn to more effectively use languages
  - Understand features and implementations
- Learn to design languages
  - Use a language to solve a problem

Can't cover everything about programming languages;
focus on some of the great features (that are repeated in many languages).

Course Motivation

How not to study programming languages concepts:

- Java in January
- Forth in February
- Modula-3 in March
- Ada in April
- ...

Course is titled “Programming Language Concepts”
and not “Programming Languages” for a reason.

Course Motivation

How we will study programming language concepts:

- use “micro” languages based on “real” languages
  - focus on the essential great features
- introduce intellectual tools to define languages
  - precision to understand great features
  - starting point for PL research

Introductions and Icebreaker

Who am I?

- Arthur Azevedo de Amorim
- Learned about PL and formal verification in college
- Went on to graduate school
  - Languages for privacy and information security
  - Formal verification of security hardware, cryptographic code, compilers, ...

Introductions and Icebreaker

Who are you?

Give your name and name a programming language (no repeats).

Course Administration

Instructor: Arthur Azevedo de Amorim

- E-mail: `aaavcs@rit.edu`
- Office: GOL-3547
- Office hours: Mo 3pm – 4pm;
  We 11am – 12pm;
  or by appointment

Website

- `http://arthuraa.net/teaching/20235/plc`
- `http://mycourses.rit.edu`

Course Rhythm

- Textbook reading
- Reading quiz
- Lectures: 1 – 3 class periods
- Recitation: 1 class period
- Programming assignment

⋮

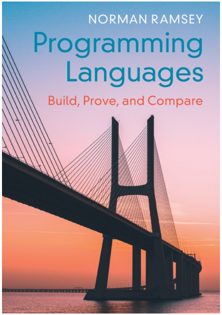
- Final Exam (Wed. May 1, 10:45am–1:15pm, SLA-2150)

Assignments, Exams, & Grades

- 10% — Reading Quizzes (≈ 13)
- 5% — Attendance & Participation (lectures)
- 5% — Attendance & Participation (recitations)
- 65% — Programming Assignments (≈ 8)
- 15% — Final exam

Textbook

Programming Language: Build, Prove, and Compare, Norman Ramsey



(Finally) Published textbook by a respected colleague.
I have used draft versions of *PL:BPC* for CSCI-344 for a number of years; feedback from previous instances of the course have been incorporated.
Available in hardcover and eBook formats.

Reading Quizzes

- on myCourses (Quizzes)
- due at 8:00AM of day of lecture that begins topic
- short (approx. 15min w/ 60min time limit) and easy (unlimited attempts), demonstrating that you have read

Attendance & Participation

Lectures:

- Won't cover everything, just the hard parts.
- Will post electronic materials to website (after lectures).
  - (but trying to use slides less and board more)

Participation means being an engaged student:

- Asking and answering questions.
- Let me know if pace is too fast or too slow.
- (Not simply attending class.)
- When I enter your grade, will I know who you are?

Attendance & Participation

Recitations:

- Small group problem solving
- Practice with key topics
- Jump start on programming assignments

Participation means being an engaged student:

- Contributing to group problem solving.
- Presenting solutions.

Programming Assignments

Demonstrate understanding of concepts and languages.

- (understanding, not mastery)

Assignments will get progressively more difficult through the semester.

Most assignments done individually; some involve pair programming.
Discussions at white board allowed and encouraged,
but do not share code.

Academic Integrity & Late Policy

Academic Integrity

- Read course policy (and linked policies)

Late Policy

- Programming Assignments due at 11:59pm
- 3 “extension tokens”
  - grants a 24-hour extension on a single programming assignment (no extensions for reading quizzes)
  - automatically applied to “Late Submissions” on myCourses (submitted after “Due Date” but before “End Date”)
  - only one extension per assignment
  - won’t answer questions about assignment after “Due Date”
- for extraordinary circumstances,
  contact an appropriate administrative staff member;
  will make special arrangements suited to the situation

Success, inside and outside of PLC

I love this material and genuinely want students to succeed at PLC.

- Aspects of the course material are meant to be challenging,
  but your time should be spent on the "interesting" parts.
- If you find yourself struggling for over an hour without making progress, then put the work aside and make a plan to seek guidance.
- Reach out via e-mail and/or attend office hours

I know that "life happens" outside of PLC.

- Make use of the many resources at RIT (incl. your instructors).
- Connect with me *early* about any difficulties you have in this course
  - Much easier to get "back on track" after a week than after three or four

What is a Programming Language?

- A “language” for writing “programs”

What is a Programming Language?

- A “language” for writing “programs”

- Language
  - Agreed upon medium for communication
  - Formal notational system
- Program
  - Computation (e.g., algorithms)
  - Executed/Interpreted by machine

What is a Programming Language?

- A “language” for writing “programs”

- Language
  - Agreed upon medium for communication
  - Formal notational system
- Program
  - Computation (e.g., algorithms)
  - Executed/Interpreted by machine

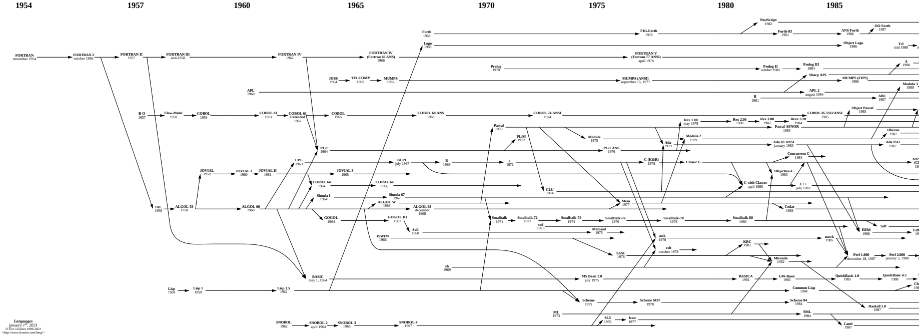
> *I like programming languages because*
> *there is nothing like a good language*
> *to help us express computations precisely,*
> *in ways that we can reason about them,*
> *while still keeping things at a high level.*
>
> *–Norman Ramsey (Tufts University)*

What is a Programming Language?

Separable concepts for defining and learning a language:

- Syntax: how do you write the various parts of the language?
- Semantics: what do programs mean?
- Idioms: how is the language used to express computations?
- Libraries: does the language provide “standard” facilities (e.g, file-access, hashtables, GUIs, . . .)?
- Tools: what is available for manipulating programs in the language? (e.g., compiler/interpreter, debugger, documenter, . . .)

History of Programming Languages



https://www.levenez.com/lang/

Programming Languages

Why are there so many different PLs?

Programming Languages

Why are there so many different PLs?

- Evolution
- Specialized problem domains
- Socio-economic factors

Programming Languages

Why are there so many different PLs?

- Evolution
- Specialized problem domains
- Socio-economic factors

Are there less PLs in 2023 than in 1989?

Programming Languages

What makes a PL successful?

Programming Languages

What makes a PL successful?

- Easy to learn
- Easy to write programs (of interest)
- Easy to implement
- Compiles to good (fast/small) code
- Socio-economic factors

- Luck?

Programming Languages

What makes a “good” programming language?

Is there a “best” programming language?

Reminders

- First reading quiz (*PL:BPC* Preface, Acknowledgements, and Introduction) due Fri. at 8:00AM