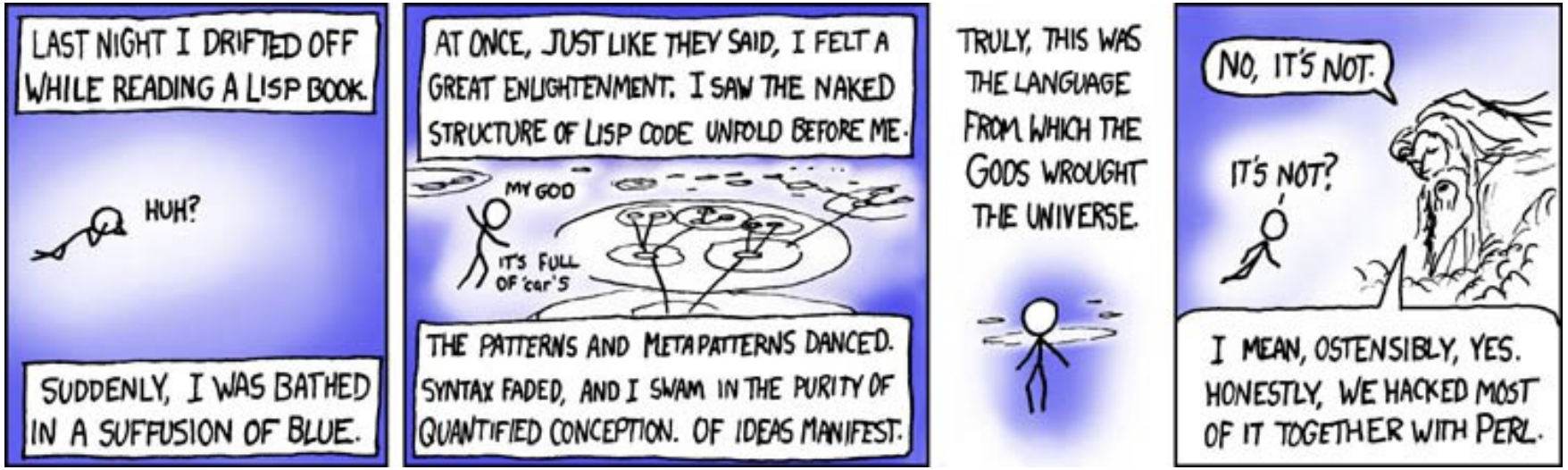
CS 430 Spring 2015

Mike Lam, Professor

Programming Languages



Today

- Course overview (what?)
- Course policies (how?)
- Course rationale (why?)

Opening Challenge

- Define "programming language"

Overview

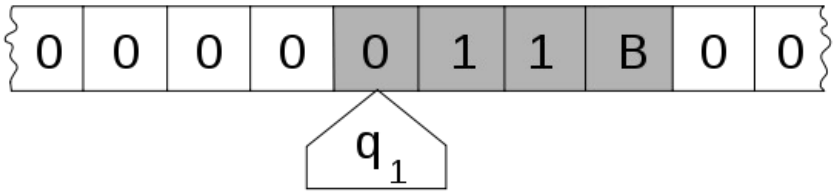
- *Programming language (PL)*
  - Tool for **formal** expression of problems and solutions
  - Audience: humans and machines
- General topics
  - **Syntax** (what a program looks like)
  - **Semantics** (what a program means)
  - **Implementation** (how a program executes)

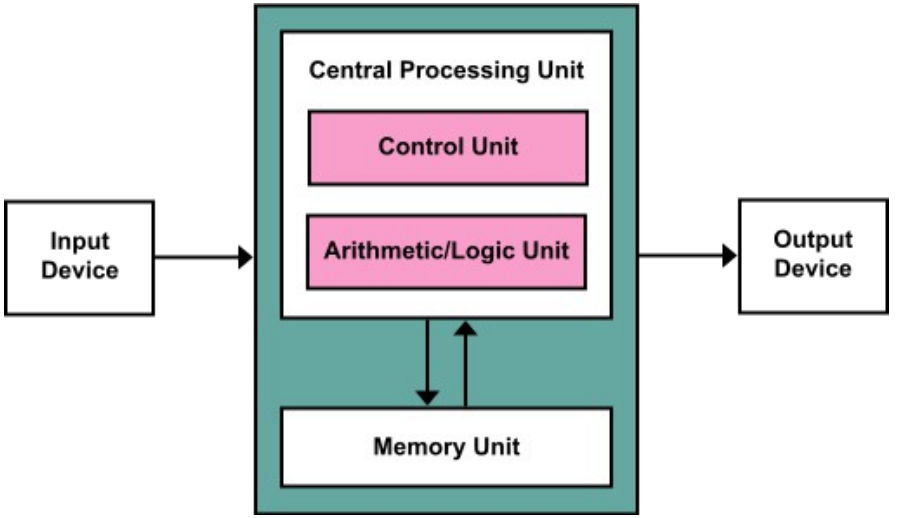
Why are PLs needed?

- Humans are excellent at approximate and contextual pattern recognition
  - This enables us to use imprecise language, which is often easier and quicker
  - Ex: "Meet you at El Charro at 7?"
  - vs. "I request your presence at 1580 S. Main St., Harrisonburg, VA, at 19:00 GMT-5 on 2015-01-13"
- Machines are less forgiving
  - They are (currently) slower and less accurate at language recognition and interpretation
  - Thus, programming in a natural language is (currently) a Bad Idea™

Every language is “special”

- Surprising result: they're (mostly) all identical from a theoretical point of view
  - A language is "Turing-complete" if it can compute anything computable by a Turing machine
  - Most modern languages are Turing-complete
- Also, (mostly) all are designed for the von Neumann architecture
  - Data and program in the same memory
  - Fetch-decode-execute cycle





Why are there so many?

- Evolution over time
  - Just like human languages
- Additionally: deliberate design efforts
  - To address shortcomings of existing languages

Which language is best?

- And why?

Which language is best?

- It depends!

Goals

- Compare programming languages with regard to syntax and semantics
- Discuss language implementation issues and the tradeoffs involved
- Gain experience in learning new languages and writing code in different language paradigms
  - E.g., scripting, functional, and logic-based

Syllabus

- It's online:
  - `http://w3.cs.jmu.edu/lam2mo/cs430/syllabus.html`
- Read it!
  - Especially the parts marked in red
- The textbook is required
  - “Concepts of Programming Languages” by Sebesta
  - Older editions should be fine
    - I do recommend the latest version
    - Watch for discrepancies re: recent languages and trends
  - No need to bring it to class

Course Website

- This is the main course website:
  - `http://w3.cs.jmu.edu/lam2mo/cs430/`
- Lots of useful stuff:
  - Syllabus
  - Calendar
  - Assignments
  - Resources
- Check it regularly!

Online Systems

- Canvas
  - Evaluations and grades
- Piazza (accessed via Canvas)
  - Q&A and discussions
- Make sure you can access all of these!

Course Policies

- Regular attendance is highly recommended
  - Watch the website for in-class quizzes and exams
  - Labs will be conducted in ISAT 248
  - If the class periods are not worth attending, tell me so that I can make them better!
- Slides will be posted on the website
  - Don't waste time writing down stuff from the slides
- Please silence your cell phones during class

Course Policies

- Submit programming projects as specified in the project description
  - No thumb drives, CDs, or emails (unless requested)
- Project grading will be based on automated test results
- Late submissions up to 72 hours will receive a 10% penalty per 24 hr period

Course Policies

- The JMU Honor Code applies on ALL assignments
  - I will use software to detect plagiarism
  - Violations may be sent to the honor council
- Unless stated otherwise on an individual assignment, all submitted code must be YOUR work entirely
  - You may work in groups to discuss assignments (in fact, I encourage this), but do NOT share code!
  - I encourage self-control: refrain from looking at others' code, no matter how casual
  - "Whiteboard rule of thumb": if you can write it on a whiteboard in under two minutes, you can probably share it with others

Class Format

- Preparation
  - Reading (textbook, links, etc.)
  - Watching (Grove's videos)
  - All units listed on website (“assignments” page)
- Reinforcement
  - Overview lectures
  - In-class activities
- Assessment
  - In-class quizzes
  - Online quizzes
  - Labs and programming assignments

Course Grades

Assignments	70%
Midterm Exam	15%
Final Exam	15%

Class Format

- This class is a hybrid of theory and practice
  - Both are important
  - Quizzes and exams will focus on theory
    - Depth (quizzes) vs. breadth (exams)
  - Labs and PAs will focus on practice
- Work load should be fairly even
  - Roughly 1-2 graded assignments per week
  - Exams are relatively low-stakes (only 15% each)
  - Check the calendar and assignment pages regularly
    - No make-up assignments
  - Do not rely on Canvas to remind you!

Course Policies

- Exams will be held in HHS 2208
- I do not curve during the semester
  - Individual assignments (usually exams) may be curved at the end of the semester
- If you ask for a re-grade, I may re-grade the entire assignment
  - This applies to homework and projects, too
- If you have to miss a due date or exam because of an excused absence, let me know ASAP
  - I do NOT guarantee make-up opportunities, but early notification certainly makes me more amenable to doing so

Contacting Me

- Questions? Try Piazza!
- Email: `lam2mo`
  - I will attempt to respond as quickly as possible, but do not expect a response in under 24 hours
- Office: ISAT 227
  - Office hours TBD
    - Please fill out the course survey!
  - Appointments preferred outside office hours

Questions?

Let's talk about PL

- Why might we want to study languages?

Why PL?

- Increased capacity to express ideas
  - E.g., use of associative maps in languages that don't explicitly provide them
- Improved background for choosing appropriate languages
  - We tend to choose things that are familiar, so it is advantageous to be familiar with many languages
- Increased ability to learn new languages
  - Practice helps, as does learning PL fundamentals
  - Also improves mastery of already-known languages

Why PL?

- Better understanding of the significance of implementation
  - Move beyond superficial differences between language syntax (whitespace, brackets, etc.)
  - Helps with program debugging
- Overall advancement of computing
  - Broader knowledge enables informed trends
  - CS does not benefit from "language ghetto" or flamewar mentalities
  - What if ALGOL 60 had become more popular than Fortran in the 1960s?

Why PL? (the real reasons)

- It looks good on your resume
- It makes you a more valuable employee
- You get to brag about all the theory and languages you know
  - We're using two "hip"/"cult" languages this semester
- It's fun!
  - (I think so, anyway...)

We already know a lot!

- Java
- C/C++
- Python
- Go
- Assembly/machine code
- Javascript
- R
- C#
- Bash
- Visual Basic
- HTML/CSS (?)

How to evaluate languages?

Evaluating Languages

- Readability
  - How easy is it to understand already-written code?
- Writability
  - How easy is it to write clear, efficient code?
- Reliability
  - How easy is it to write programs that adhere to specifications?

Evaluating Languages

- Simplicity
- Orthogonality
- Data types
- Syntax design
- Support for abstraction
- Expressivity
- Type checking
- Exception handling
- Restricted aliasing
- Standardization

Evaluating Languages

- Simplicity (few basic constructs, minimal overloading)
- Orthogonality (independence of features, feature symmetry)
- Data types (expressive without being redundant)
- Syntax design (consistency, sensible keywords)
- Support for abstraction (subprograms, data structures)
- Expressivity (conveniency, "elegance")
- Type checking (strict is safer, but cost vs. benefit is debatable)
- Exception handling (early detection, clean handling)
- Restricted aliasing (make it apparent)
- Standardization (respected organization, appropriate time)

Evaluating Languages

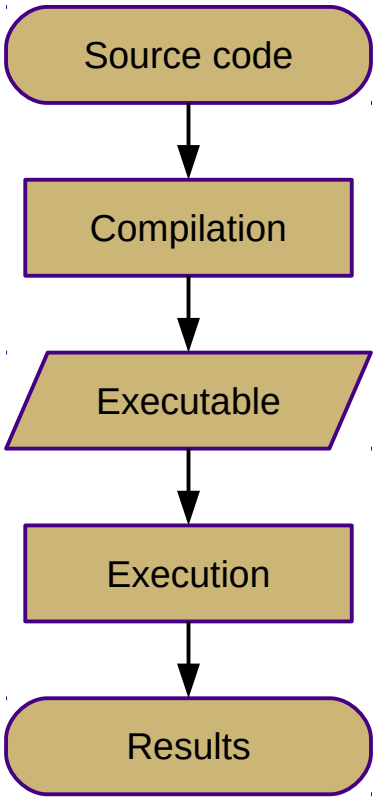
- Various costs
  - Programmer training
  - Code writing and debugging
  - Compile time
  - Execution time
  - Runtime system
  - Maintenance
  - Porting
- Tradeoffs exist between these costs
  - Language designs represent points on these spectrums

Language Categories

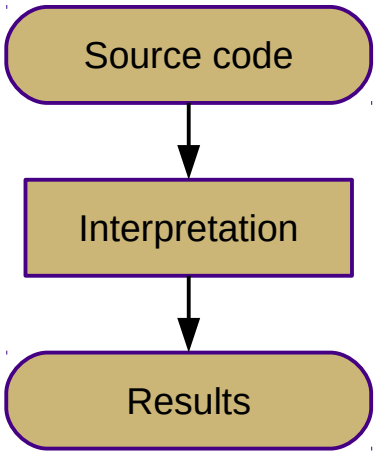
- Traditional bins:
  - Procedural/imperative (assembly, Fortran, COBOL, ALGOL, C)
  - Functional (Lisp, Scheme, Haskell)
  - Logic- or rule-based (Prolog)
  - Object-oriented (Smalltalk, C++, Java)
- Other bins:
  - Visual (Visual Basic, Adobe Flash)
  - Scripting (Perl, Javascript, Python, Ruby)
  - Markup or metadata (HTML, LaTeX)
  - Educational (Scratch)
  - Special-purpose or domain-specific (DSL)

Compilation vs. Interpretation

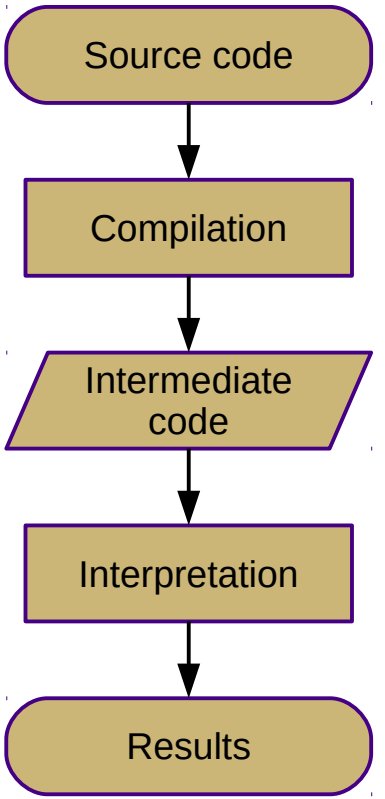
Compiled



Interpreted



Hybrid



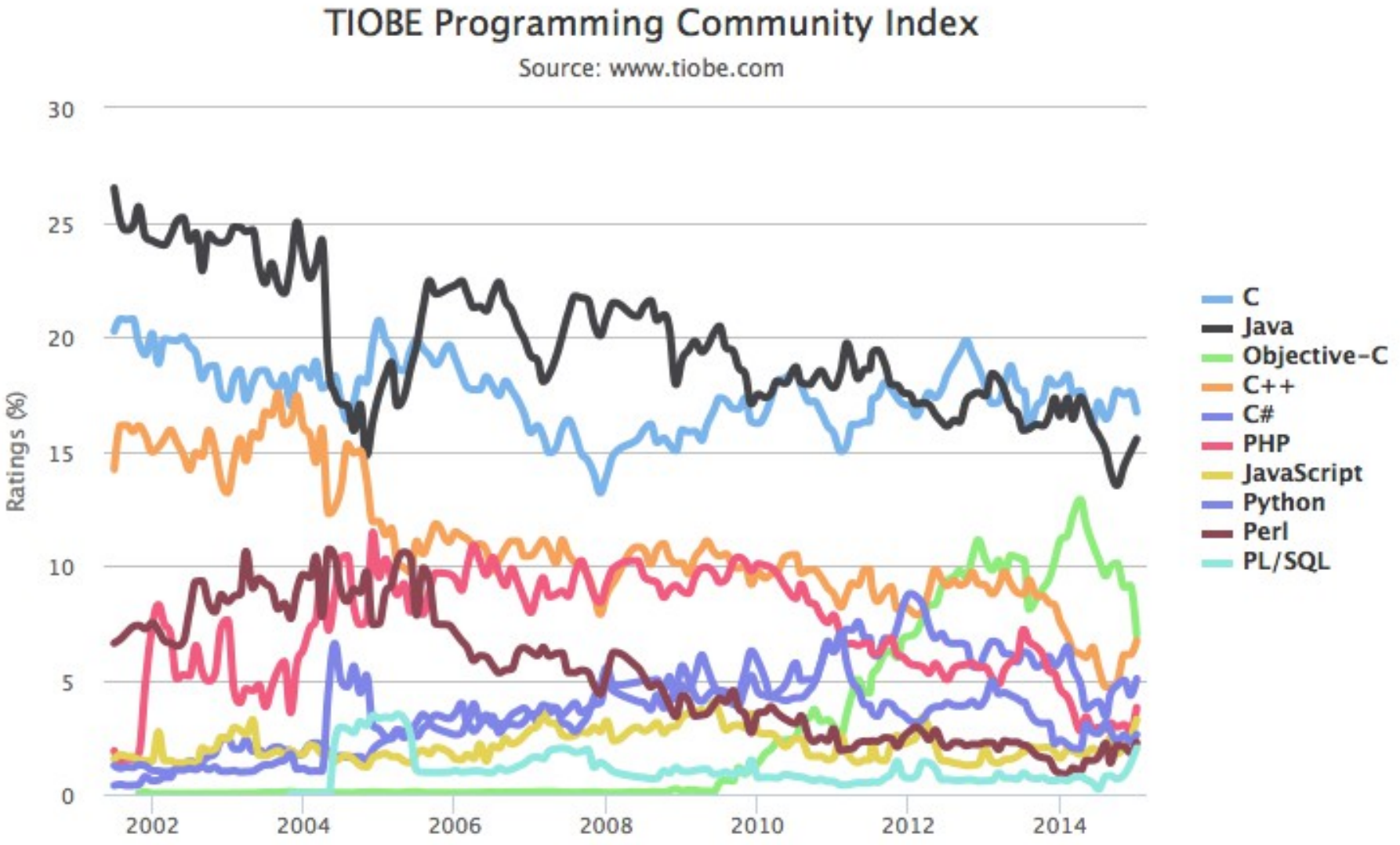
Context: Programming Domains

- Scientific
  - Primary concern: efficiency (speed)
- Business
  - Primary concern: data processing and formatting
- Artificial intelligence
  - Primary concern: symbolic computation
- Systems
  - Primary concern: efficiency, low-level access, and portability
  - Value of language safety is hotly debated
- Web
  - Primary concern: presentation and ease of development

Context: PL Design Influences

- Hardware/architecture design shifts
  - Historic prevalence of imperative/procedural languages that closely match the hardware
  - Cheaper hardware → higher-level languages
- Software development methodology shifts
  - Better software engineering practices and a call for safer languages
  - Agile programming and rapid prototyping languages
- Social, cultural, and political shifts
  - Millennial and post-millennial generation culture ("hip" web languages and software systems)

Relative Popularity



Historical Popularity

Programming Language	2015	2010	2005	2000	1995	1990	1985
C	1	2	1	1	2	1	1
Java	2	1	2	3	-	-	-
Objective-C	3	23	39	-	-	-	-
C++	4	3	3	2	1	2	12
C#	5	5	8	8	-	-	-
PHP	6	4	4	30	-	-	-
Python	7	6	6	24	22	-	-
JavaScript	8	8	9	7	-	-	-
Perl	9	7	5	4	9	19	-
Visual Basic .NET	10	-	-	-	-	-	-
Pascal	16	13	77	12	3	20	5
Lisp	18	16	12	15	5	3	2
Ada	30	26	15	16	6	4	3

Primary Course Languages

- Ruby
  - Latest version: 2.2.0
  - ISAT 248 lab version: 2.0.0
- Haskell (GHC)
  - Latest version: 2014.2.0.0
  - ISAT 248 lab version: (same)
- Prolog (GNU)
  - Latest version: 1.4.4
  - ISAT 248 lab version: (same)
- All of these can be installed on your local machine
  - I recommend that you do so!
  - GNU/Linux and Mac OS X are fine
  - Windows can be difficult (I recommend Cygwin or a virtual machine)

Homework

- Complete course survey by Thursday (on Canvas)
- Read Chapters 1 & 2
  - Optionally, watch videos on PL history
- On Thursday
  - Lecture/discussion on Unit 2 (PL history)
  - Quiz on Unit 1 (Introduction)

Good luck!

- Have a great semester!