Introduction to C

CS 2022: Introduction to C

Instructor: Hussam Abu-Libdeh
Presented by: Renato Paes Leme

Cornell University

Fall 2009, Lecture 1

Administrivia

- Instructor: Hussam Abu-Libdeh, 4139 Upson
- Email: hussam@cs.cornell.edu
- Lectures: MWF 12:20–1:10p, THR 205
- Lab: Upson B7 (when announced)
- Office Hours: To Be Determined
- `www.cs.cornell.edu/courses/cs2022/`
- CMS: `http://cms.csuglab.cornell.edu/`

C is a language...

C is a language...

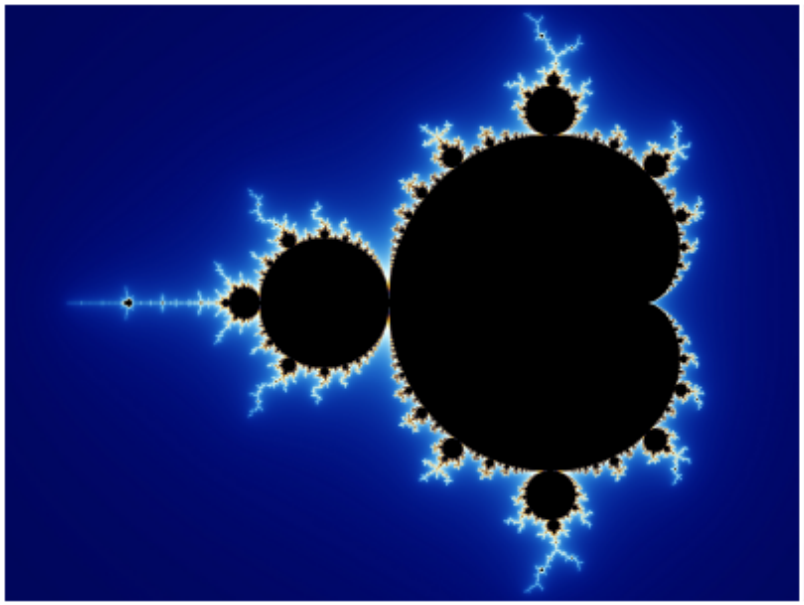
..., like English.
What are languages for?

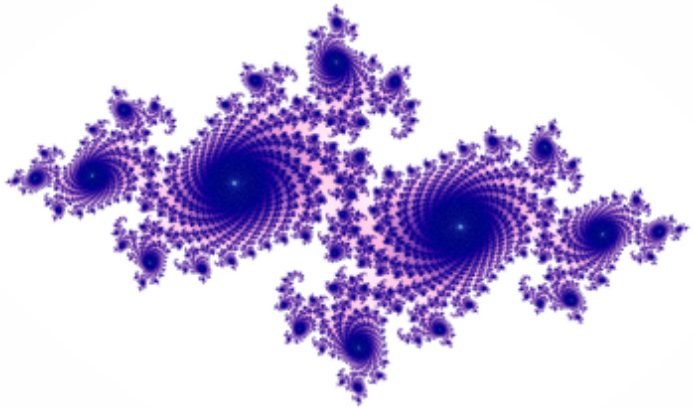
C is a language...

..., like English.
What are languages for?

- communicate ideas
- convince someone
- express feelings and beliefs
- vehicle for art
- state ideas
- ...

Visual art

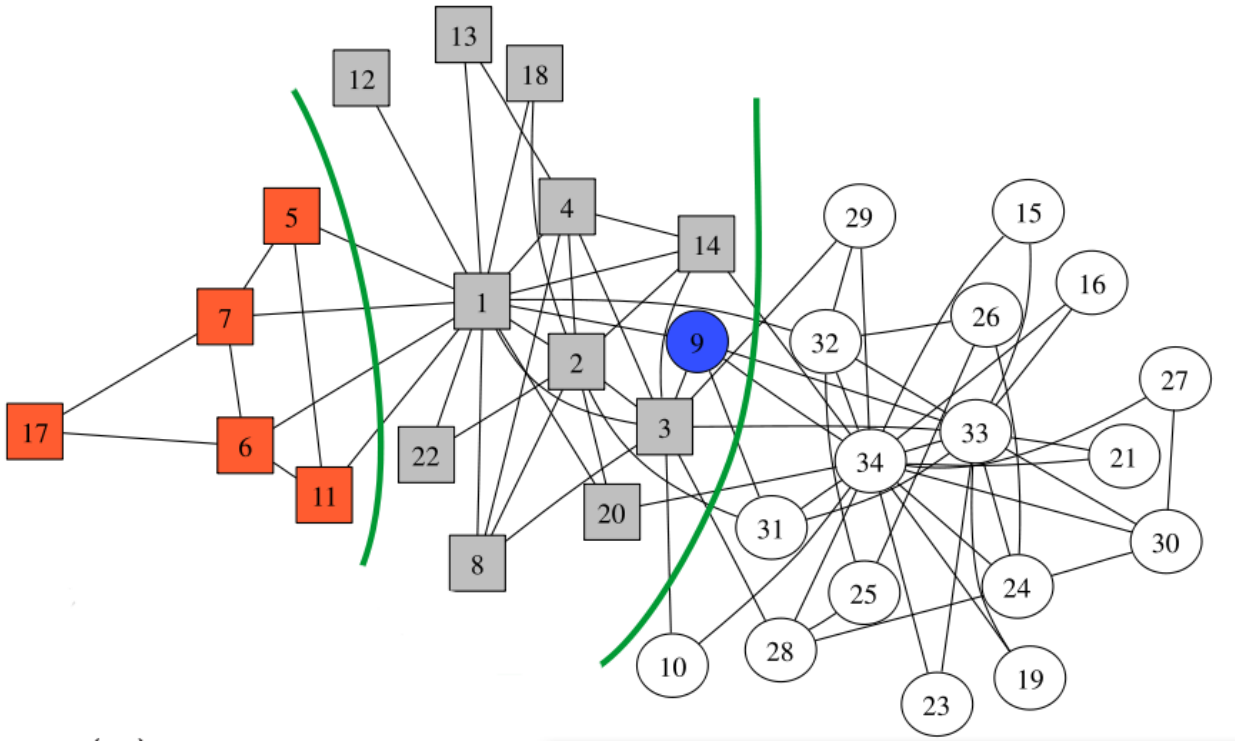




Computer generated poetry

Imagine now a tree in white sails still whirled
About the leaves
will be of silences
Calm and angels

Social Sciences



Zachary's karate club

Communicate mathematical ideas

1. Simulations, e.g., Monty Hall Problem
2. Solve large systems of equations
3. Automatic provers
4. ...

Design useful tools

Google™

How to learn a foreign language?

1. be able to express yourself
2. have someone to correct your grammar
3. develop some style
4. communicate efficiently
5. say original and interesting things

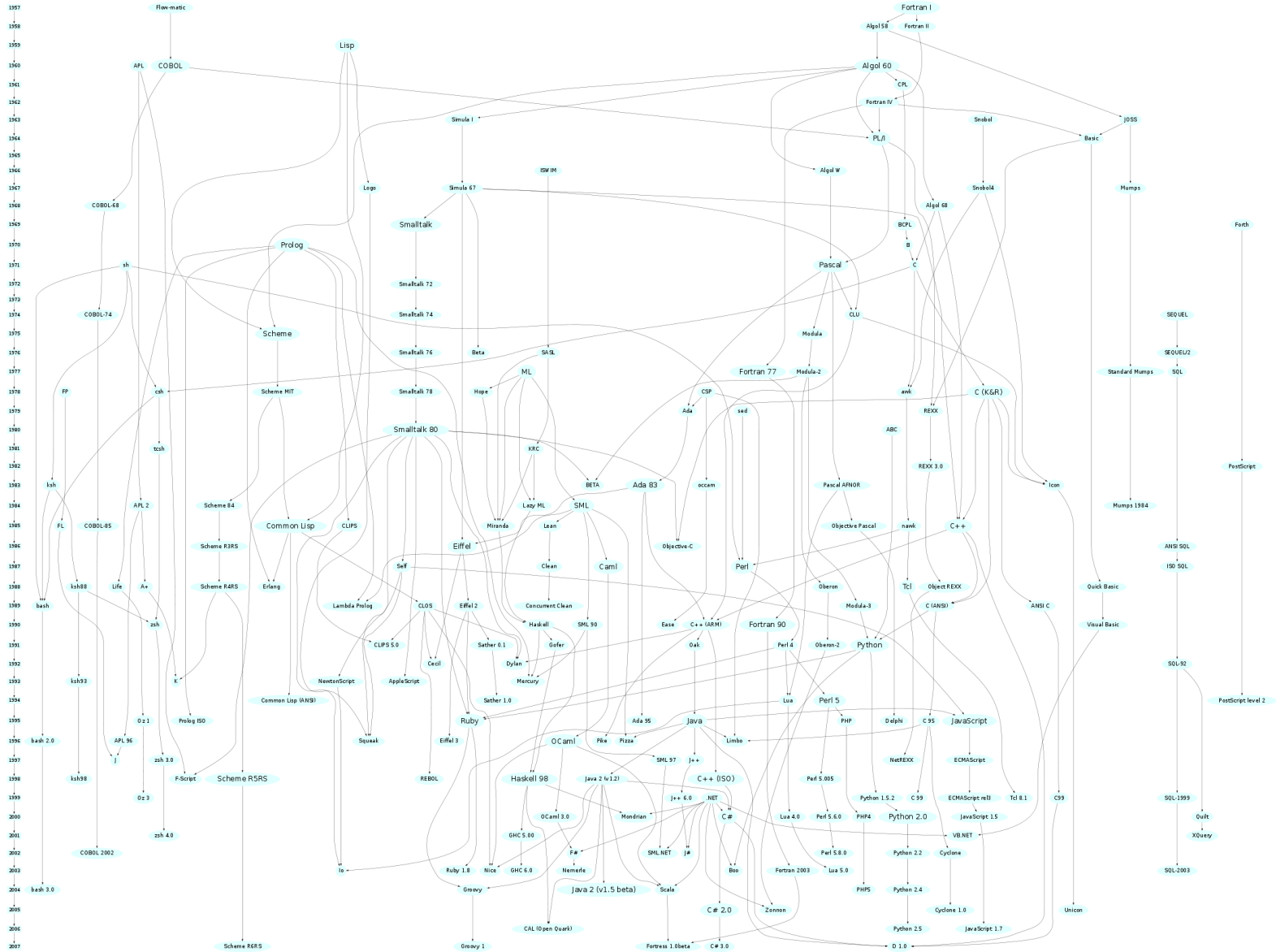
How to learn a programming language?

1. express what you want in C
2. let the compiler correct you
3. know good coding practices
4. optimize your code
5. write beautiful code

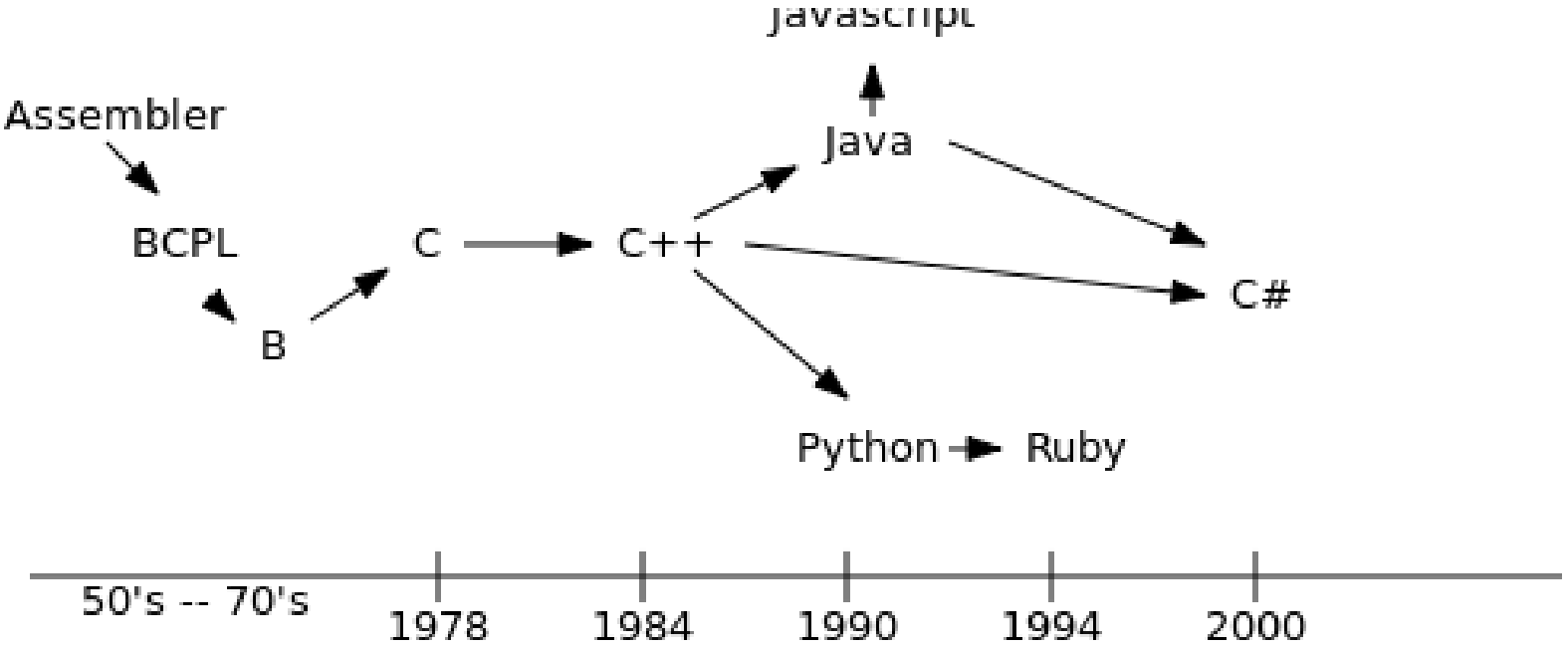
Pre-requisites

- Basic programming knowledge (variables, functions, loops)
- Lots of composure
  - Your programs won't compile
  - Your programs won't run
  - Your programs will crash
  - You'll have no idea what happened
  - ... but at least it'll happen fast!

Programming Languages Genealogy



Programming Languages Genealogy



History of C

- Writing code in an assembler gets real old real fast
  - Really low level (no loops, functions, if-then-else)
  - Not portable (different for each architecture)
- BCPL (by Martin Richards): Grandparent of C
  - Close to the machine
  - Procedures, Expressions, Statements, Pointers, ...
- B (by Ken Thompson): Parent of C
  - Simplified BCPL
  - Some types (int, char)

History of C

- C (by Kernighan and Ritchie)
  - Much faster than B
  - Arrays, Structures, more types
- Standardization
- Portability enhanced
- Parent of Objective C, Concurrent C, C*, C++

When to use C

- Working close to hardware
  - Operating System
  - Device Drivers
- Need to violate type-safety
  - Pack and unpack bytes
  - Inline assembly
- Cannot tolerate overheads
  - No garbage collector
  - No array bounds check
  - No memory initialization
  - No exceptions

When not to use C

Use JAVA or C# for ...

- Quick prototype
- Compile-once Run-Everywhere
- Reliability is critical, and performance is secondary
  - C can be very reliable, but requires tremendous programmer discipline
  - For many programs, JAVA can match C performance, but not always

How to be good programmer

How to be good programmer

- Practice, practice, practice

How to be good programmer

- Practice, practice, practice
- Code for fun

How to be good programmer

- Practice, practice, practice
- Code for fun
- Don't wait. Google for it.

How to be good programmer

- Practice, practice, practice
- Code for fun
- Don't wait. Google for it.
- Coding competitions:
  1. USACO: `train.usaco.org`
  2. Topcoder: `www.topcoder.com/tc`
  3. UVA: `uva.onlinejudge.org/`

Questions?