Lecture 01: How to Pick a Language?
August 24th, 2017

Announcements

• Expect PA1 to be handed out Tuesday
  – Due 1 week from Tuesday (midnight)
  – Expect this as a general pattern
• Office Hours:
  – David White (lab)
    • M 1:00-3:00,
    • Th: 1:00-3:00
  – Jason Yu (lab)
    • M: 3:00-5:00
    • Tu 8:00-9:00 (AM), 2:00-3:00
  – Bruce Draper (CSB 442)
    • Th 11:00-12:00,
    • F 9:00-10:00
• Recitation conflicts
  – Want to switch recitations?
    • My hands are tied: recitations are full
    • Exception: Thursday morning – you can transfer into that one
  – Or: find someone who wants to switch with you
    • How? I suggest people who want to switch meet by the podium after class

Review: Graded Class Components

• Quizzes (9%)
• Recitations (11%)
• Midterms (20%; 10% each)
• Final Exam (20%)
• Programming Assignments (40%; 4% each)

To get a grade of C or better, you must have a total weighted score of 70 or better AND scores of 65 or better in both the programming assignments and the tests (midterms & final)

Review: Programming

• Ten programming assignments
  – Each builds on and/or modifies the previous
  – Practice living with large code
  – One will be graded on efficiency
    • Premature optimization is the root of all evil (Knuth)
• This is your last programming class!
  – Future classes cover topics
    • Security, Theory, Graphics, Compilers …
  – All assume you can design, implement, test & maintain large programs

Review: Academic Integrity

• All graded work in this course is individual
• And because of the law of large numbers
  – Likely someone here now will try to cheat.
• We will actively look for all such students.
  – … and impose punishments (see guidelines).
• Also, in cheating, giving is as bad as taking.
• Question: “Can’t I talk with my classmates?”
  – Yes you can and you should:
    • exam study groups, concepts, etc.
  – You can’t design/write code for/with them.
Review: Exceptions

- Deadlines are deadlines, except in the case of unforeseeable emergencies
- For example
  - Death in the family
  - Illness requiring hospitalization
  - House fire
- In such cases, talk to the instructor

Weiss, Chapter 0

- You’ve read it (and taken the quiz)
- The title was “Introduction”
- But what was the point of this chapter?

Weiss Chapter 0: Ten Reasons to Use C++

1. C++ is still widely used
2. Templates
3. Operator overloading
4. Standard Template Library
5. Automatic reclamation of resources
6. Conditional compilation
7. Distinctions between accessors and mutators
8. Multiple inheritance
9. Space efficiency
10. Private inheritance

How to pick a programming language

How to pick a language?

- Choose a language designed for a similar purpose
  - Languages are complex ecosystems
- C was designed to write operating systems
- C++ was designed to add OO to C
  - Inherits OS goals from C
  - New goal: large software systems (objects)
- Java was designed to write web applications

Reasons to Program in C++

- To interface with legacy code
  - C, C++, or Assembly
- To directly interact with hardware
  - Device drivers, OSs, etc.
- To optimize machine efficiency
  - Time & space
  - Even at the expense of programmer efficiency
- Real-time performance
  - Defined as the ability to write code that will finish in a given amount of time
  - E.g. self-driving cars
Reasons to Program in Java

• To optimize programmer efficiency
  – Sometimes at the expense of machine efficiency
• For portability
  – Execute the same way on any machine
• For security
  – Java sandbox makes foreign code safer

Friction:
C++ vs Java Goals

• Direct hardware access vs Security
• Direct hardware access vs Portability
• Machine Efficiency vs Portability
• Real-time Performance vs Programmer Efficiency
• Legacy code vs Portability

Weiss’ High-level Differences Redux

1. Security & robustness: absolutely
   – Security is a goal of Java, not C++
     • References (Java) vs. Pointers (C++)
       – C++ also has references...
     – Robustness has multiple meanings
       • Robustness to bugs
         – Part of programmer efficiency (a goal of Java)
         – Fewer choices, fewer mistakes
       • Robustness to hacks
         – Part of security (again, a goal of Java)

2. Compiled vs interpreted code
   – Java is “interpreted”
     • Compiles to Java VM code
     • Java VM interprets it one line at a time
     – Goal: portability
   – C++ is “compiled”
     • Compiled & linked straight to machine code
     • Directly executed by OS (no VM)
     – Goal: machine efficiency

3. Multi-threading
   – Many types of parallelism
     • Processes vs threads vs SIMD vs ...
   – Java includes multi-threading
     • Implemented in the Java VM
     • Goal: portable parallelism
   – C++ does not include any form of parallelism
     • Platform specific libraries support various models
     • Goal: machine efficiency (pick best model)