Today

• Integration

Homework

• HW9 out (due Tues)
Integration Concepts

Integration:
- combining separate software components into a single system
- interwined with the construction sequence (order parts are developed)

Daily Build & Smoke Tests
- "Build" means create an "executable" ... make your product "runnable"
  - What building means depends the type of software you are building
- "Smoke tests" are simple checks to see if product "smokes" when run

Q: Advantages of Build & Smoke Tests?
- easy way to find basic integration issues
- ensures program can always be "build" and "run"
- isolates issues
  - e.g., if it could build properly on Day 17 but not on Day 18
  - problem isolated to code added between two builds
- improves moral — seeing your product build and run is motivating!
- surfaces needed work and issues that can be addressed early
  - as opposed to unexpectedly at the end of the project
  - helps avoid the end-of-project "tar pit"
  - amortizes work throughout project instead of at very end
  - don’t want to fight with integration issues as deadline approaches!
Smoke Tests: Exercise entire system from end-to-end

- but not exhaustive tests
- just enough to expose major problems
- if passes suggests system is ready for more detailed testing
- smoke test has to evolve as system grows

Exercise: What is a good “smoke” test for your end-of-semester prototype?

Exercise: What about for your final project?

A “good” build minimally should ...

- compile all files, libraries, and other components successfully
- link all files, libraries, and other components successfully
- not contain any showstopper bugs (pass the smoke test)

Exercise: What will a “good” build be for your project?
Best Practice: Automate your daily build & smoke tests ...

Build automation tools

- make (long live make!)
- ant, maven, gradle, ... (Java)
- webpack, gulp, grunt, ... (Javascript)
- Travis CI, Jenkins, ... (continuous integration)

Smoke test tools

- basic automation test frameworks (e.g., unit tests)
- UI test frameworks (e.g., selenium, XCode, Cypress/Jest, ...)

Exercise: Spend a few minutes and research possible build and (smoke) test tools for your project