CS263 Goals

Learn and understand the various core functions of modern runtime systems

Gain hands-on experience with a modern distributed runtime system (cloud platform)
  - Understand the web-based software architecture

Dig deeper into the software stack that ultimately executes the program
  - Managed runtime system, aka (programming language) virtual machine
  - We’ll focus on Java (but I’ll point out the diffs with other languages as appropriate)

Programmer productivity aids that impact performance (and how we try to recover it)
  - Portability of apps
  - Garbage collection
CS263 Course Topics

- Software, services, and their interoperation
  - Google App Engine – course project overview and setup
- Modern languages and their managed runtime systems
  - OO review
  - Implementations
  - Java focus
- Garbage collection
- Execution: interpretation, dynamic/JIT compilation
- Adaptive (profile-driven) optimization
CS263 Course Topics

- Software, services, and their interoperation
  - Google App Engine
- Modern languages and their managed runtime systems
  - OO review
  - Implementations
  - Java focus
- Garbage collection
- Execution: interpretation, dynamic/JIT compilation
- Adaptive (profile-driven) optimization
CS263 Evaluation

http://www.cs.ucsb.edu/~cs263/

- 10% Class attendance and participation
  - Read papers, come prepared to ask/answer questions on the topic
- 20% Assignments (on schedule), quizzes in and out of class
- 40% Midterm
  - Review in class prior, midterm in class (closed notes/books/etc)
- 30% Project (1-2 person groups)
  - Weekly code commits (starting with Assignment 3)
    - Public github repo
  - Youtube presentation/demo at end
CS263 Project

30% of class grade (1-2 person groups)

- Java Google App Engine App (free tier)
  - Required APIs: datastore (no JPO/JDA/objectify), task queue, cloud storage, and memcache
  - Must expose a GET and POST REST API for each GAE API/service
    - REST APIs must use JSON to send/receive data
    - Turnin includes curl commands that work on Linux to read (GET) and write (POST) each service
    - Use of JAXRS
  - Use of maven to package and build the application
  - Selenium or other tool script that exercises app APIs, with usage in project README
- Public github repo, fully documented, commit (100+ lines/wk) required over time incl. tests
- 15-20 minute public, recorded (youtube, vimeo, ...) demo and presentation
  - Overviewing project, contributions, novelty, experience, performance evaluation, and demonstrating app execution
- **Turnin**: link to github repo, link to running app, link to video, file with tested curl commands
  - App to be kept running until graded (Friday of finals week)
CS263 Fall 2014: Questions?

- Instructor: Chandra Krintz
  - HFH 2153
  - Office hours by appointment, skype (ckrintz), chat (ckrintz@gmail.com, ckrinintz)

  - Lectures posted (slides and youtube)

- Class starts promptly at 9am (please be on time)
  - Will end between 10:15 and 10:45 depending on the topic
  - Assigned readings on website/schedule should be read by the class date indicated