

software studio

intro & logistics

Daniel Jackson

goals: what you'll learn

design skills

- › problem analysis
- › behavior design
- › design tradeoffs

programming skills

- › reflective coding
- › new languages & frameworks

professional practice

- › team work
- › shared repositories
- › presentation

course elements

solo projects in phases

- › phase 1: get going
- › phase 2: minimal viable product
- › phase 3: adding cool features
- › phase 4: critique & peer review

team project

- › you choose the team and the project

class sessions

- › nuggets and review of your work
- › readings for each class, with nanoquiz

recitation

- › focused on technology

grading policy

components

- › solo 40%, team 30%, participation & quizzes 10%
- › excellence 20%: best design and code work

collaboration on solo projects

- › can discuss plans & work, but must name your collaborators
- › all written work must be your own

lateness

- › 3 slack days; no other lateness
- › free slack days for illness with medical note

class contract

The teaching staff of the course will endeavor to:

- › Deliver content that is stimulating, practically useful and of long term intellectual value;
- › Encourage interaction during all class and recitation sections, and take all questions seriously;
- › Provide constructive and helpful feedback on student work in a timely fashion;
- › Assign grades in a fair and consistent way;
- › Take seriously suggestions about course content and structure;
- › Help students who feel overwhelmed or lacking in background, despite prerequisites;
- › Support and encourage full participation of all students.

As a student, by taking this course, you agree to endeavor to:

- › Attend all class sessions, and to participate enthusiastically;
- › Not use laptops and mobile devices during class for email and other distracting activities;
- › Share concerns and comments about the course with the lecturer;
- › Help create a sense of community by making forum postings anonymous only when absolutely necessary;
- › Do your fair share of work on your team project.

why X?

why Ruby?

- › Rails: lets you avoid SQL
- › strong culture of elegance & simplicity
- › cleaner syntax, more uniform than Python
- › good to learn another language

why GitHub?

- › most popular open source host
- › good support for issue tracking too
- › they gave us free private repos (thank you!)

your turn

lab hours

- › installation clinic: tonight 5-7pm

read website

- › esp. general info & solo project overview

recitation tomorrow

- › go to any recitation

P0: due Monday

- › Hello World in Rails
- › stored in GitHub, deployed to Heroku

MIT OpenCourseWare
<http://ocw.mit.edu>

6.170 Software Studio
Spring 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.