

Lecture 03: Planning Your Course and Creating Awesome Assignments

CS298 / EDUC298

Spring 2021

Stanford University

Computer Science Department

Lecturer: Chris Gregg

[PDF of this presentation](#)



Lecture 03: Planning Your Course

- What do you want your students to be able to **do**?
- Don't re-invent the wheel! Do a targeted search for other, similar courses.
- Writing a syllabus
 - A good start:
 - Course logistics (e.g., meeting location and times)
 - Prerequisites
 - Textbook and other material requirements
 - Topics the course will cover, with or without the schedule
 - Course assignments
 - Exam dates
 - Grading information
 - Course policies, expectations, and guidelines
 - Honor code and/or plagiarism information
- Planning the schedule
 - To start, broadly place the topics you want to teach in lecture spots
 - You may not know specifics initially, but do the best you can
 - Put down assignments -- when they will be released and due.
 - When are the exams going to be? Will this affect assignment due dates, etc.?



Lecture 03: Planning Your Course (continued)

- Plan your website
 - Have a template, or use a Learning Management System (e.g., Canvas)
- Are you going to hire TAs?
 - Where will you get them
 - Will they have taken this course or a similar one before?
 - If you can find one or two key TAs to help you with the course, this will be extremely helpful
- Other administrative details
 - Will you need lab space?
 - What kind of classroom / lecture hall do you want?
- Start early!
 - It takes a significant amount of time to plan a course. Don't try to do it in the two weeks before class begins.
 - If nothing else, have the first week (or two!) planned exquisitely before the course begins.



Lecture 03: Creating Awesome Assignments

- What was your favorite assignment ever?
 - What made it awesome?
 - Interesting assignments are the ones your students will remember!
- Creating an awesome assignment
 - Get ideas from other sources -- e.g., <http://nifty.stanford.edu>
 - What do you want your students to be able to **do**?
 - Match the course content to the assignments.
 - Early assignments are often easier (though not always)
 - You probably won't get every single topic in the course into an assignment, and that is okay!
 - Decide whether you want to build assignments off of other assignments or not
 - What are you going to do if students don't finish the first assignment correctly?
- Students love assignments that have a real-world feel
 - Add context if you can! Is there a historical angle you can leverage? Is there a current-events topic that is relevant (e.g., security issues at Company X?)
 - Motivate the assignment in class or in the assignment itself
- Be careful with games -- some students love them, others hate them. Try not to have more than one or maybe two game assignments per quarter.
- Students like to be challenged! Make your assignments doable but challenging.



Lecture 03: Creating Awesome Assignments (continued)

- Scaffolding your assignments
 - Are you going to have starter code?
 - Will students be expected to understand a lot of code before they begin the assignment?
 - You can tailor the assignment based on the amount of starter code, but this is tricky
- Writing assignment instructions
 - Your instructions should have:
 - Assignment deadline and how to submit the assignment.
 - Startercode download link, or instructions on how to setup the assignment (e.g., if it is on a server).
 - A description of the assignment, with motivation for why the assignment is interesting or important. Background information about the problem itself, with an example of what a student's solution should produce or answer, can also be an important part of the description.
 - Further examples of output, if relevant, and instructions on how students can test their solutions.
 - Detailed implementation instructions (or not -- this depends on the assignment!)



Assignment 3 – Due Tuesday, April 22nd, in class

- Planning Your Course:
 - If you have an idea for a course you would like to teach, describe it in broad strokes, e.g.
 - What is the *core* idea of the course?
 - What is the importance of the course?
 - Where does the course fit into the CS curriculum?
 - Who is your targeted audience?
 - What kinds of assignments will you have in the course?
 - What other assessments will you give? (e.g., exams, papers, etc.?)
 - If you don't have an idea for a course, find a course you are currently taking, and analyze the course from the perspective of how it was designed. See if you can answer the questions above based on the course. If it isn't clear from the syllabus or course information handouts, did the instructor talk about them anywhere else?
- Creating Awesome Assignments:
 - If you have a course in mind, describe one assignment that you want in the course but have not yet planned.
 - Where in the curriculum will the assignment fall?
 - Could it move somewhere else if necessary (e.g., does it require material later in the course that can't be moved forward?)
 - What will the students be able to **do** by completing the assignments (e.g., what the goals?)
 - What is going to make the assignment awesome?
 - How will you grade the assignment?
 - Describe the best assignment you've ever done
 - Why was it awesome?
 - What did you enjoy the most about the assignment?
 - Could you make it better in some way?

