

Lecture 02: Know Your Students, Know Your Subject, Know Your Tools

CS298 / EDUC298

Spring 2021

Stanford University

Computer Science Department

Lecturer: Chris Gregg



[PDF of this presentation](#)

Lecture 02: Know your Students

- When teaching a course, you should understand your students abilities, and you should teach the class at the appropriate level for those students.
 - Too many instructors don't teach their course at the right level, and students are either underprepared, or end up with a course that does not challenge them.
- At the beginning of the course, it helps to get any information from the students that might be important:
 - What do they want to be called? (knowing as many names as possible is a good thing, but difficult!)
 - Where are they in their academic career? (e.g., Freshman, Sophomore, grad student, etc.)
 - Do they have any concerns about the course?
 - Why are they taking this course?
 - What is their background with the material, or with the prerequisites?
 - Do they need any accommodations?
 - Do they have a particularly busy schedule this term?
- You will meet students before and after lecture, and during office hours or other meetings -- get to know them then! Ask about how the course is going, how assignments are going, and what their general perception of the course is. This is fantastic feedback!
- You can find out the perception of a course before you teach it, if it is something that has been taught before. E.g., CS107 is known to be a hard class, so that is nice to know up front.
- Teaching CS1 is particularly challenging because students can come in with a wide range of abilities.
- What do you do if a student tells you something concerning?



Lecture 02: Know Your Subject

- Presumably, you know a bit about computer science already!
 - Teaching a subject is inherently different than knowing it, but the more knowledgeable you are about your subject, the better you can teach it.
 - You *will* learn things about your subject you didn't know before you taught it.
 - The better you know your subject, the easier it will be to predict questions, and to give more detail when answering questions. Your overall lectures and assignments will likely be more robust, as well.
 - Sometimes, being an expert makes it *more* difficult to teach a subject -- you forget what it is like to not understand something, which seems obvious now.
- If you end up having to teach a course that you haven't taken before, or that you are very rusty on:
 - There is a plethora of online materials for virtually any subject.
 - Don't discount reading a textbook or two
 - You can lobby to take a course (online or in person)
 - See about co-teaching first
- Teaching a course multiple times is awesome. It allows you to make changes the second time through -- the first time through, things won't go perfectly.
- Sometimes you may think some students know the material better than you do (and sometimes, that is true!)
 - Don't discourage the students, and try to use their knowledge to help others
 - If a student always seems to be answering all the questions, make a conscious effort to allow others the time to answer.



Lecture 03: Know Your Tools

- There is an amazing number of tools available to help make your class better, particularly if you have large classes
 - Computer scientists love to create tools, and there are probably more tools available for CS than any other subject
 - Don't re-invent the wheel if you can help it: there may already be the exact tool you need, which you may be able to use for free, or cheaply
 - Your university may have certain tools that they purchased (e.g., Canvas)
 - Sometimes, you may need to write your own tools (and sometimes you just want to do it).
- Types of tools you may want to use
 - Assignment submission tools
 - Autograders
 - Assignment grading beyond autograders (e.g., TAs can add comments, etc.)
 - Test grading software
 - Test taking software
 - Student feedback software / websites
 - CS demonstration software
 - We will cover this in more detail later in the course, but there is a phenomenal assortment of tools available online that demonstrate CS concepts
 - Gradebook software (or Excel)
 - Online forums (Piazza, Slack, etc.)
 - Your course website



Assignment 2 – Due Thursday, April 15th, in class

- Know Your Students:
 - What is something a teacher (or teachers) of yours did to get to know the students in class?
 - If you have a large (> 40) class, what are some ideas of how to get to know them?
- Know Your Subject:
 - Your department chair tells you that you are going to teach a course you have never taken nor taught before. What are your strategies to do the best you can with the course?
- Know Your Tools:
 - You assign your CS1 students the following assignment:
 - Write a Python program that draws text rectangles, given two parameters on the command line.
 - E.g., `python rectangle.py 4 7` draws the following rectangle:
- If either of the parameters is missing, the students should print:
`usage: python rectangle.py width height`
 - If either of the parameters is not an integer, the students should print an error (text up to them. This error should be handled first).
 - If either of the parameters is negative or 0, the students should print:
 - **Both sides of a rectangle must be positive!**
- See next slide for solution program and assignment.

```
myth51$ python rectangle 4 7
****
*  *
*  *
*  *
*  *
*  *
*  *
*  *
****
myth51$
```



Assignment 2 (continued) – Due Tuesday, April 16th, in class

- Know Your Tools (continued)
 - Here is a working solution:

```
#!/usr/bin/env python3

import sys

def drawRec(w,h):
    SYMBOL = "*"
    if w <= 0 or h <= 0:
        print("Both sides of a rectangle must be positive!")
        return
    topAndBottom = SYMBOL * w
    middle = SYMBOL + ' ' * (w-2) + SYMBOL
    print(topAndBottom)
    for i in range(h-2):
        print(middle)
    print(topAndBottom)

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: {} width height".format(sys.argv[0]))
        quit()
    try:
        drawRec(int(sys.argv[1]),int(sys.argv[2]))
    except ValueError:
        print("Both parameters must be positive integers.")
```

- Assignment:
 - How might you go about designing an autograder for this CS1 assignment?
 - What would your autograder have to be able to do?
 - Are there corner cases in your autograder?
 - Be as detailed as you'd like about how your autograder should work.
- Optional:
 - *Build* an autograder for this assignment, in any language you choose.

