

Innovative Instruction in the CS Classroom

Why faculty aren't obsolete ... yet

Kate Lockwood

Cal State, Monterey Bay



Overview

- Background on how I got interested in innovative ways to teach CS
- Introduction to some instructional techniques
 - Inverted classroom
 - Peer instruction
- Discussion: opportunities in CS education

My Background



Bachelor's in Computer Engineering

University of Michigan: 43,000 students, Est. 1817



Master's in Information Science

University of Michigan: 43,000 students, Est. 1817



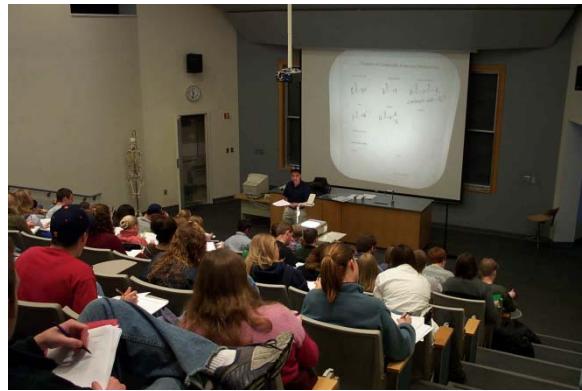
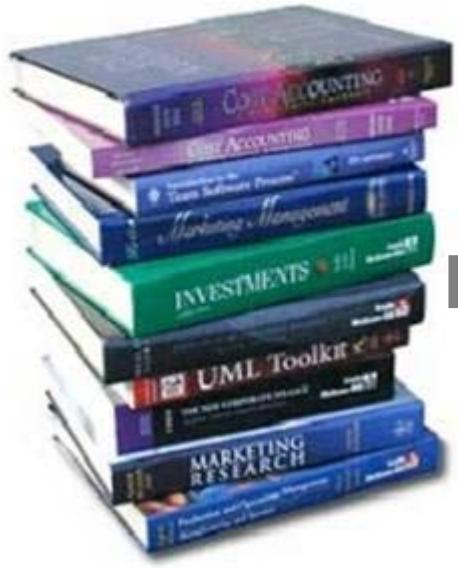
PhD in Computer Science

Northwestern University: 20,000 students, Est. 1851

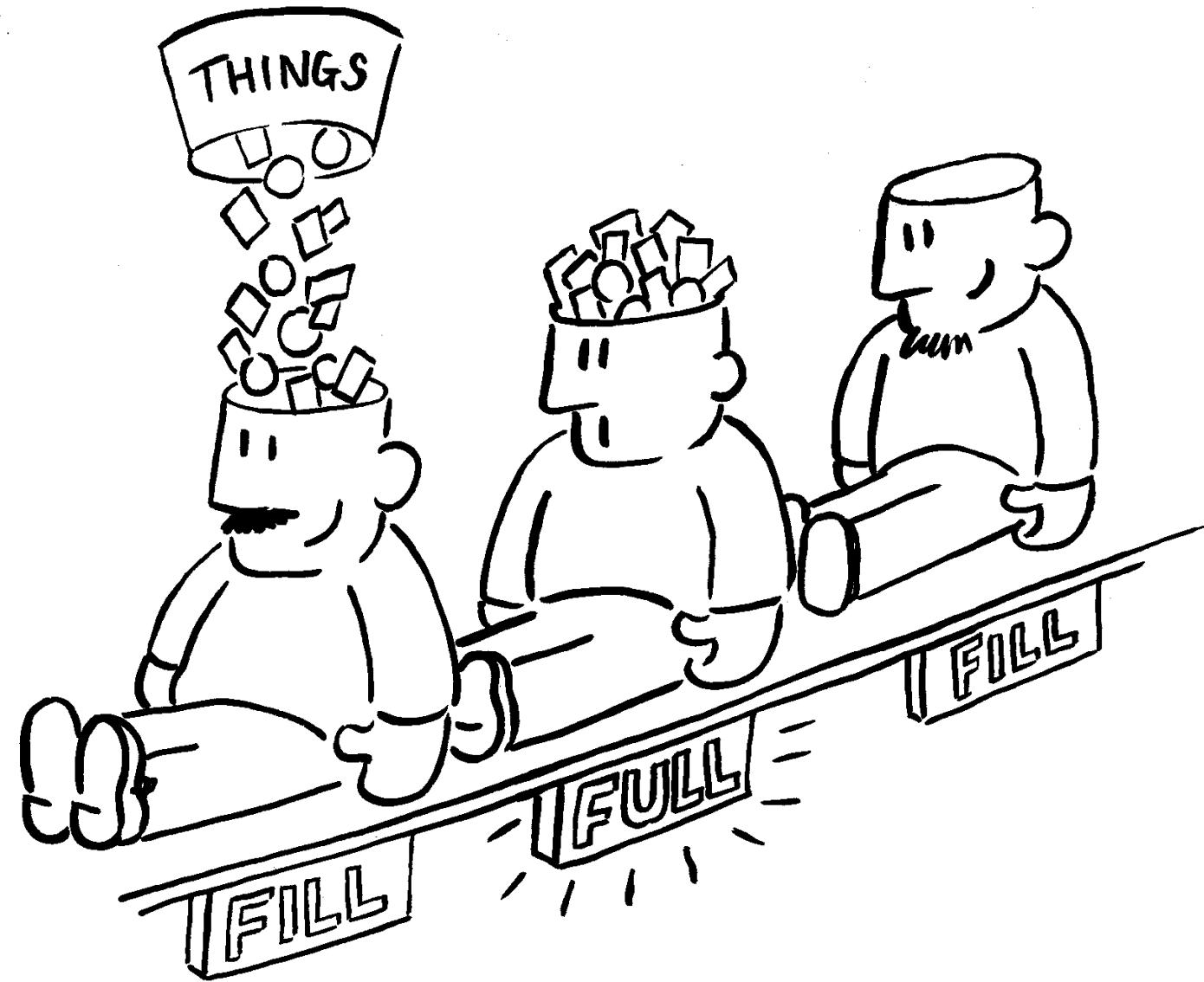


Assistant Professor – 4 years

Cal State Monterey Bay: 5,000 students, Est. 1994



LEARNING







Unengaged students

High absence rates

Copied homework

Consistent fail rate
~25%

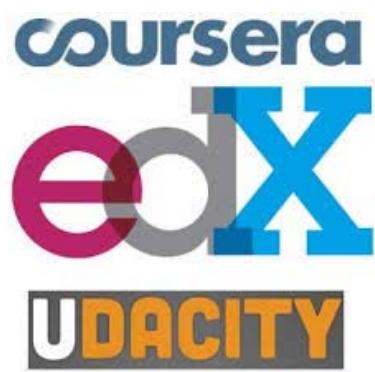


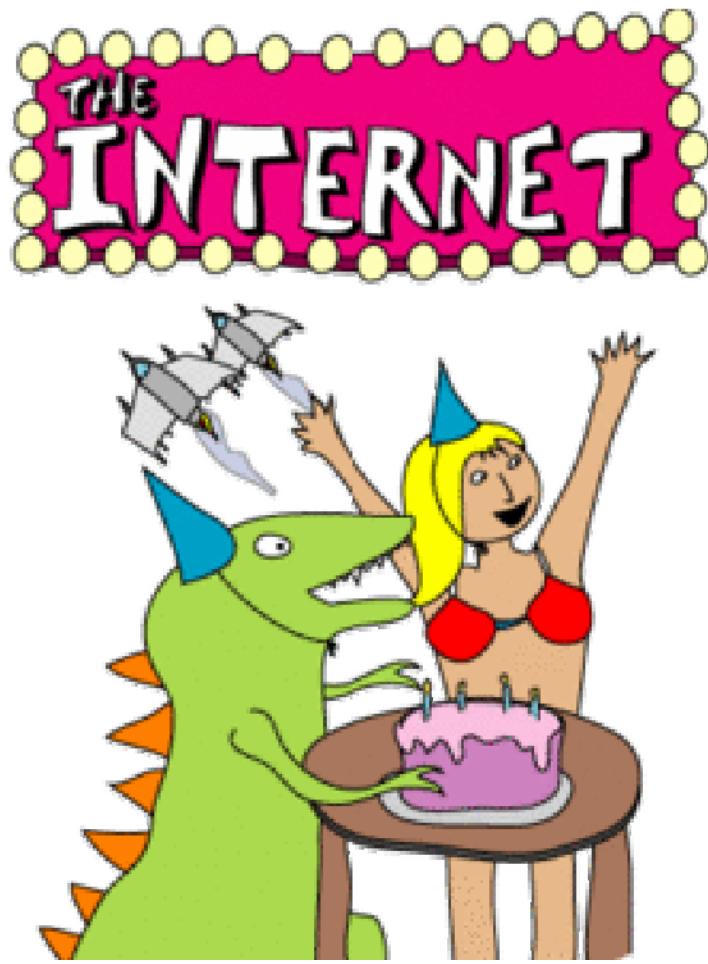
“I’m just not good at computer science”

“I understand the concepts but can’t figure out the programming problems”



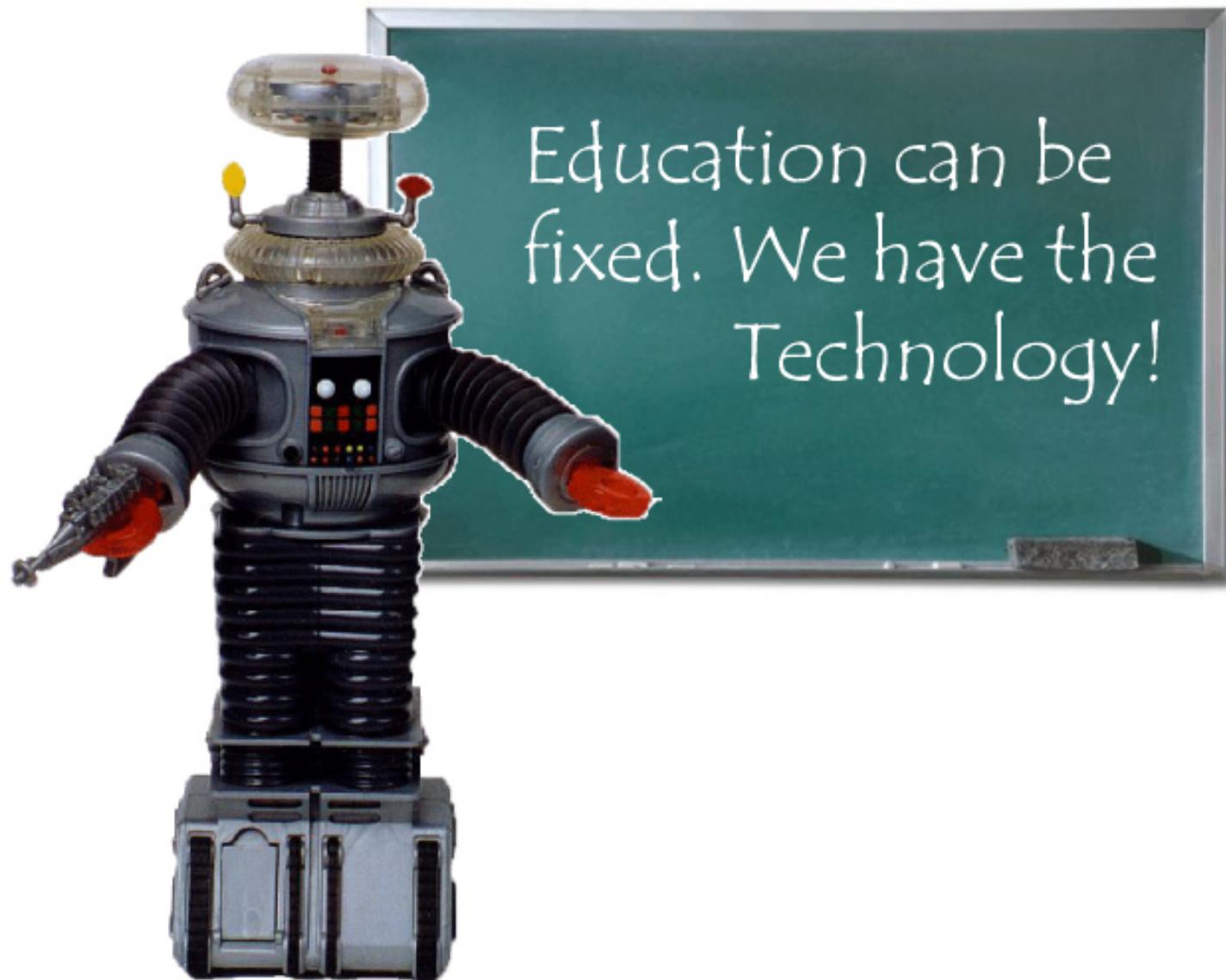
the sage on the stage





CLASS
VS.

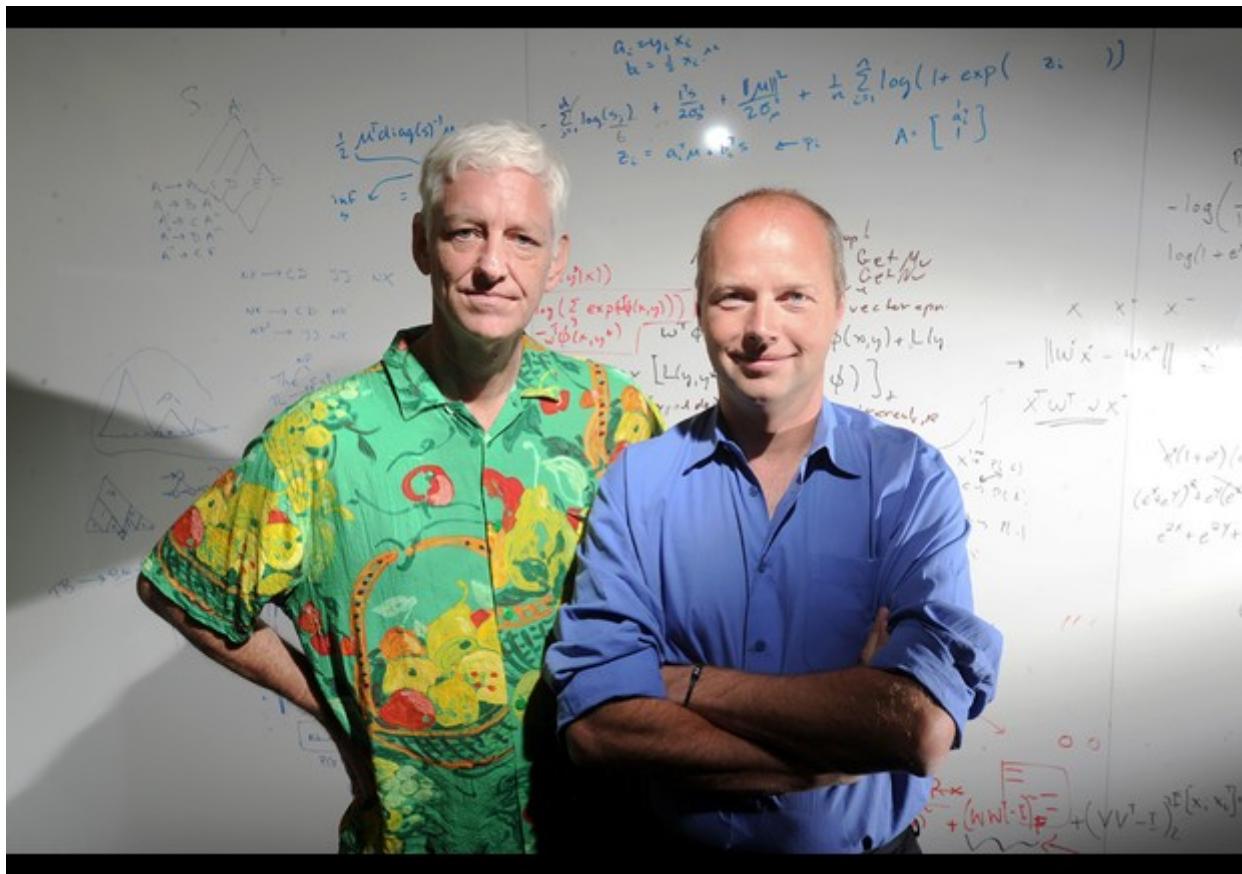




Education can be
fixed. We have the
Technology!

RISE OF THE MOOCS





MOOCS



8 HOURS per week spent
on a MOOC while in session



33,000 STUDENTS enrolled per class



- “...in 50 years there will be only ten universities left in the world”

ACM TECHNEWS

A Master's-Level Computer Science Degree, Delivered Via MOOCs

ZDNet

May 22, 2013

Comments

VIEW AS: SHARE:



The Georgia Institute of Technology, College of Computing, plans to offer the first online Master of Science degree in computer science that can be earned via a massive open online course (MOOC) format. The degree will be delivered through the Udacity MOOC platform, and AT&T will provide financial support.

Students enrolled in the program will pay a fraction of the cost of traditional on-campus master's programs. Total tuition for the

SIGN IN for Full Access

User Name

Password

[» Forget Password?](#)

[» Create an ACM Web Account](#)

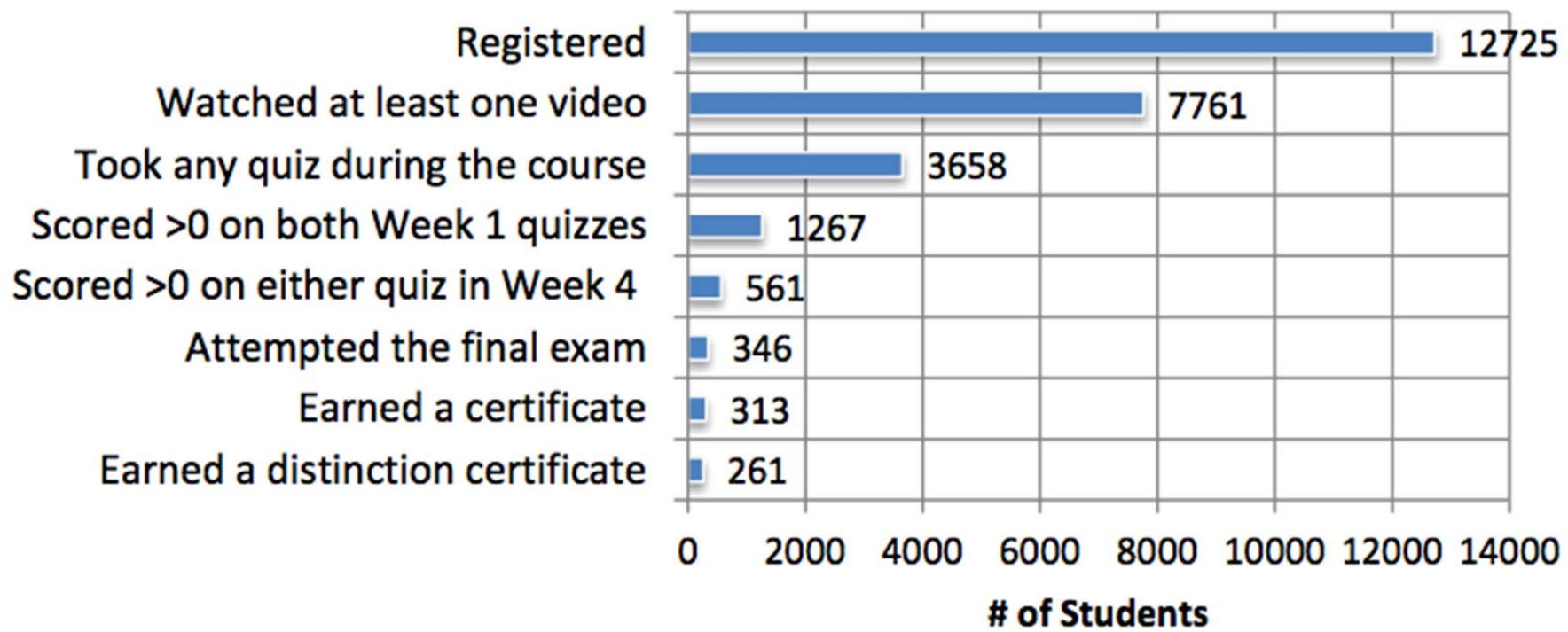
SIGN IN

MORE NEWS & OPINIONS

[Improving Communication](#)

By Michael S. Goosman

Figure 3. Student persistence in *Bioelectricity, Fall 2012*



Coursera Partnership

- *The partnership with Coursera will give professors the option to experiment with and improve upon the “blended learning” model, which combines online video lectures and content with active, in-person classroom interactions.*

How do students learn?

Acquisition



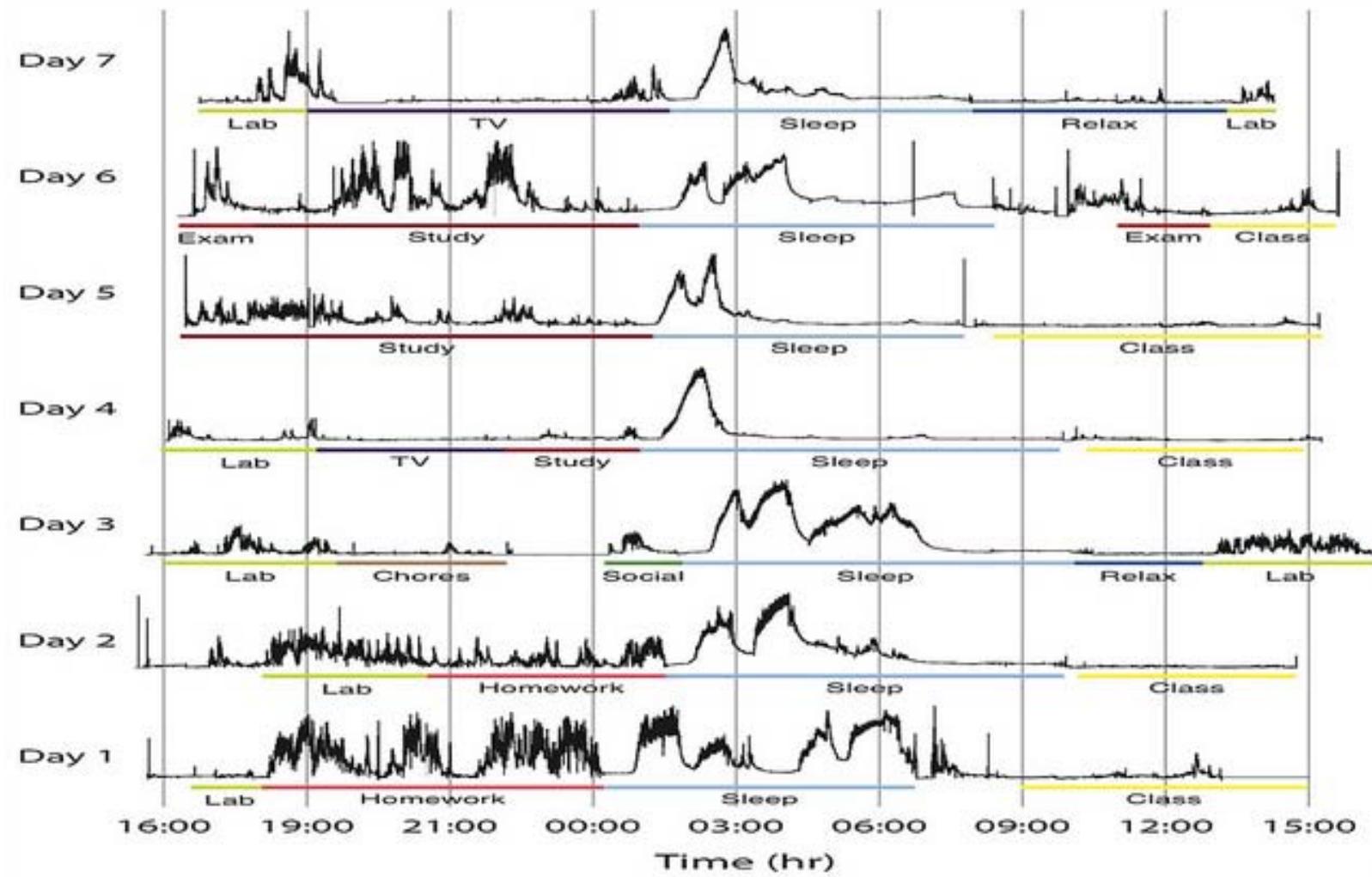
Lectures, readings,
videos

Assimilation



Solving problems, hands-on projects, creative works





Lecture Course

- Lecture
- Quizzes
- Exams

*Acquisition &
Demonstration*

Class



- Reading
- Problem sets
- Projects

Mostly Assimilation

Home





Inverted Classroom

The Inverted Classroom



CS20 at Harvard

- **Homework would be daily.** There would be a reading assignment for every class. But when they got to class, **they would talk to each other instead of listening to me.** In class, I would become a **coach** helping students practice rather than an oracle spouting truths. We would “flip the classroom,” as they say: students would prepare for class in their rooms, and would spend their classroom time doing what we usually call “homework”—solving problems.
- And they would **solve problems collaboratively**, sitting around tables in small groups. Students would learn to learn from each other, and the professor would stop acting as though his job was to train people to sit alone and think until they came up with answers. A principal objective of the course would be not just to teach the material but to persuade these budding computer scientists that they *could* learn it.

Inverted Classroom

- Lecture
- Quizzes
- Exams

*Acquisition &
Demonstration*

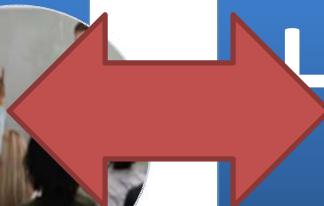
Class



- Reading
- Problem sets
- Projects

Mostly Assimilation

Home



Pilot Course: CST 231

Homework

- Fewer projects
- Online workbooks

for loops

Just about any kind of repetition you could want to model can be done with the while loop that we learned last time. In C++ for loops are also handy for modeling repetition, they are particularly good at writing loops that involve a counter (or doing an action a known number of times). The syntax for a for loop is a little more involved and in the general case, it looks like this:

```
for(initial statement; loop condition; update statement)
{
    statement(s);
}
```

The rules for how a for loop

1. The initial statement is
2. The loop condition is
3. The statement(s) in the loop will update the loop

TO DO: How many times does the loop below print "Looping"?

- Watch: [int main\(\)](#)
- Watch: [int num=10, sum = 0;](#)
- Down: [for\(int i=1; i<=num; i+=3\)](#)
- Click: [cout << "Looping!\n";](#)

Your answer

Submit

for loops

```
int main()
{
    int num, sum = 0;
    cout << "How many times would you like to loop? ";
    cin >> num;
    for(int i=1; i<=num; i++)
    {
        sum += i;
        cout << "Your final sum = " << sum << endl;
    }
}
```

$$\begin{array}{c|c|c} \text{num} & \text{sum} & \\ \hline 3 & 0 & 1 \\ & 0+1=1 & 2 \\ & 1+2=3 & \\ & & \vdots \end{array}$$

In-Class

- Quizzes
- Group problem solving
- Labs
 - Done with pair programming partners

Pilot Course: CST 205

- Use existing resources when possible
- Free online books
- Codacademy
- Media computation

The screenshot shows a computer screen displaying a Python coding environment on the Codecademy website. The top navigation bar includes links for 'Create Account' and 'Sign In'. The main content area is titled 'How to Think Like a Computer Scientist' with a logo featuring a green snake and a question mark. Below the title, it says 'Learning with Python', '2nd Edition (Using Python 2.x)', 'by Jeffrey Elkner, Allen B. Downey, and Chris Meyers', and 'Last Updated: 21 April 2012'. The central part of the screen shows a code editor with the following Python code:

```
1 # Set the variable赋值给变量
2   name = "Always look on the bright side of life!"
```

To the right of the code editor is a large, dark rectangular box labeled 'Kone'.

Step One: Strings
This course assumes familiarity with the material presented in Lesson 1: Python Syntax.
Another useful data type in Python is the string. Strings are, well, strings of characters, which is a more formal way of saying they're really just regular English phrases. They can include numbers, letters, and various symbols. For an example, [click here](#).

A string literal is a string created by simply just writing it down between quotation marks (" ") or (' '). You have to use the same type of quotation mark on each end of the string, though—the string is not ["string"](#).

INSTRUCTIONS
Assign the string "Always look on the bright side of life!" to the variable `name`.

Way to go! Start Next Lesson

**Do they do the
work??**

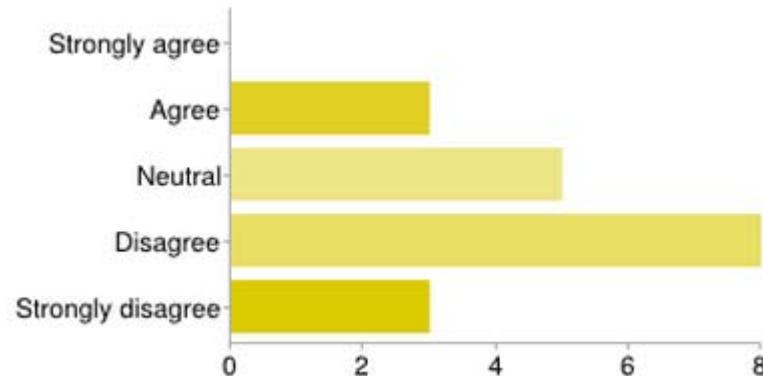
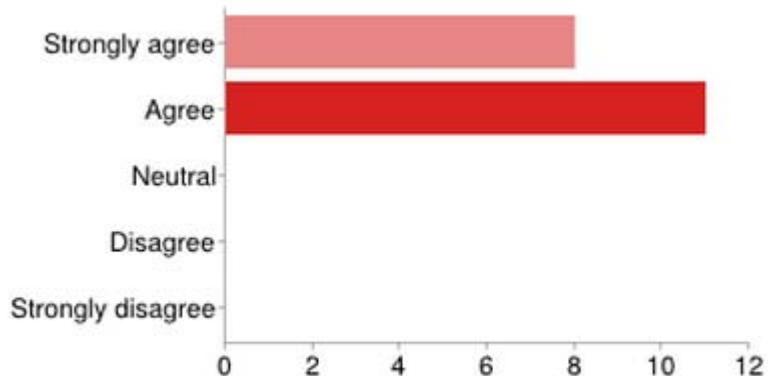
How often did you complete the workbooks before coming to class?



How often did you review a workbook after its due date to reinforce a concept or for clarification during lab?



- I like the hands-on problem solving work in class
- I would prefer a lecture-based course



Peer Instruction

Peer Instruction

Pre-class preparatory work

Question posed to class (typically multiple choice,
often using clickers for student response)

Students discuss answers in small groups

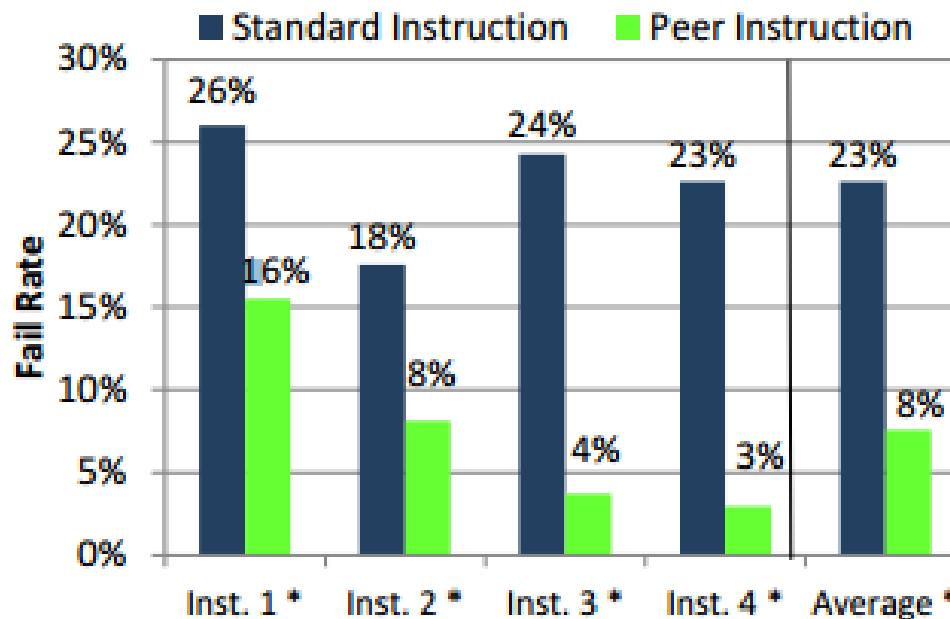
Question answered again (students may change
their answer based on group discussion)

Class-wide discussion led by the instructor

Example Question

- Which of the following is best suited for a dictionary instead of a list?
 - A. The order in which people finish a race
 - B. The ingredients for a recipe
 - C. The names of world countries and their capital cities
 - D. 50 random integers

Peer Instruction



- Porter, L., Bailey-Lee, C., and Simon, B. (2013). *Halving Fail Rates Using Peer Instruction: A Study of Four Computer Science Courses*. SIGCSE 2013.

Existing Resources

PEER INSTRUCTION
FOR COMPUTER SCIENCE

About Advice All Courses Latest Research

OPERATING SYSTEMS

Operating Systems Peer Instruction Materials

Published July 21, 2012 - No Comments

Topics Coverage Summary: This is an introductory course on the principles of operating systems. Topics include processes, scheduling, synchronization, memory management, virtual memory, file system, I/O, protection, security, networking, and distributed systems.

There is a significant sys ...

[read more](#)

◀ CS1 IN PYTHON

CS 2 IN JAVA

OPERATING SYSTEMS

CS1 IN MATLAB

CS PRINCIPLES
with Alice

COMPUTER ARCHITECTURE

Other Innovations/Techniques

- Problem-based learning
- Pair programming
- Peer review





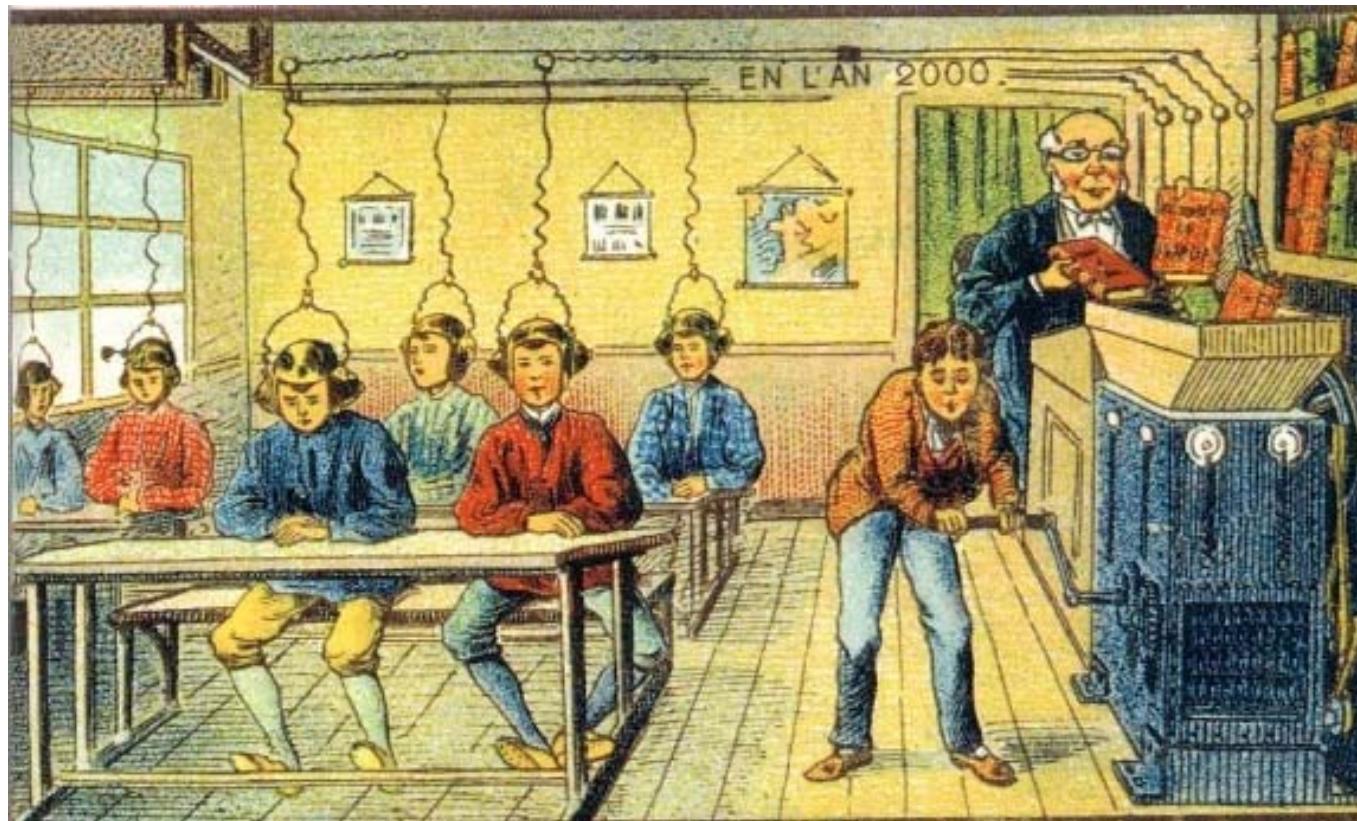
A collage of various programming language names and their descriptions, including Logo, LabVIEW, Perl, PHP, SQL, .NET, XML, C++, Pascal, Functional, Scala, VisualBasic, MATLAB, Cobol, Yorick, Assembly, Fortran, Prolog, Scheme, Python, Haskell, JavaScript, C, C++, C#, Smalltalk, Basic, BASIC, Interpreted, Distributed, Logic-based, Markup, Script, Object-Oriented, Prototypical, Declarative, Ruby, Erlang, Data Structured, Scripting, Java, Multicore, Multiparadigm, and Mjölk.



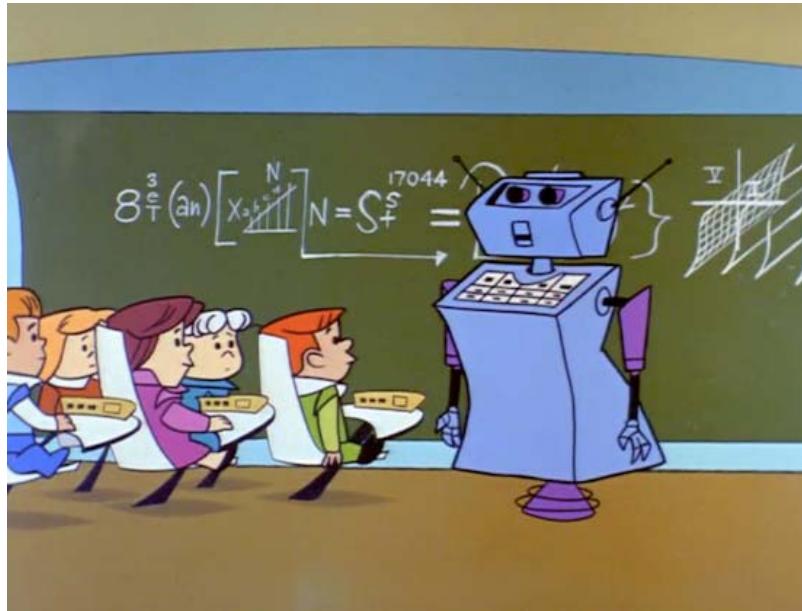
Some Commonalities of Successful Innovations

- Frequent, low-stakes, formative assessments
- Some materials provided for home-study
 - Students held responsible for materials
- Hands-on, in class problem solving
- COLLABORATION

Education != Content Delivery



What (I think) I've Learned



- Instructors still have a valuable role to play
- Encourage students to learn through ***doing***
- If you assign homework, hold students accountable
- Use technology as a means not an end
- Borrow liberally from others

Questions/Discussion