



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

Dan Garcia, Ph.D.

Lecturer SOE, EECS Dept, UC Berkeley
(on Sabbatical in Melbourne until 2008)

www.cs.berkeley.edu/~ddgarcia/



Overview

Quickie background
SIGCSE plug

This talk from 3 in 2007
ACE2008 in Wollongong

Improving CS education



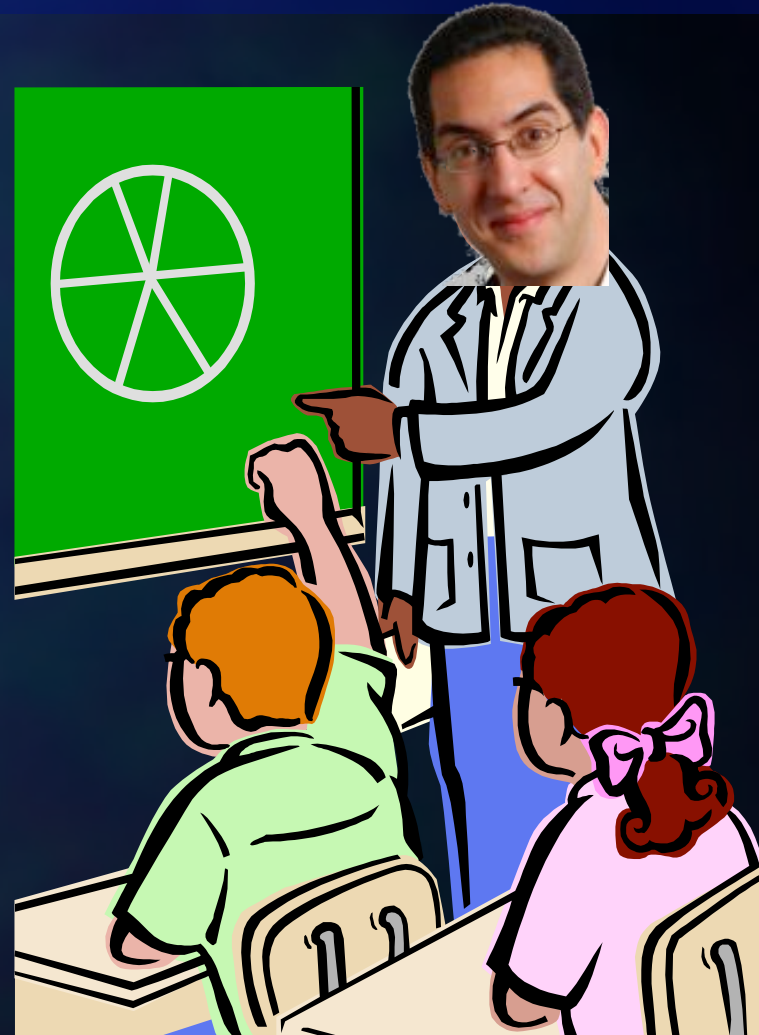
Teaching Tips



Best Practices from
survey of US schools



ACM Education Board





Teaching Tips Archive

www.cs.berkeley.edu/~ddgarcia/teaching/tips/

Goal: Gather great CS teaching tips on a wiki

4 seasoned educators seeded with 10 tips each

Categories : Lecturing, Office (hrs), Staff, Exams, Labs, Section, Projects, Grading, Meta



Dan Garcia
UC Berkeley



Owen Astrachan
Duke Univ



Nick Parlante
Stanford Univ



Stuart Reges
U Washington



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

3/24



Share Your CS Passion (Lecture)

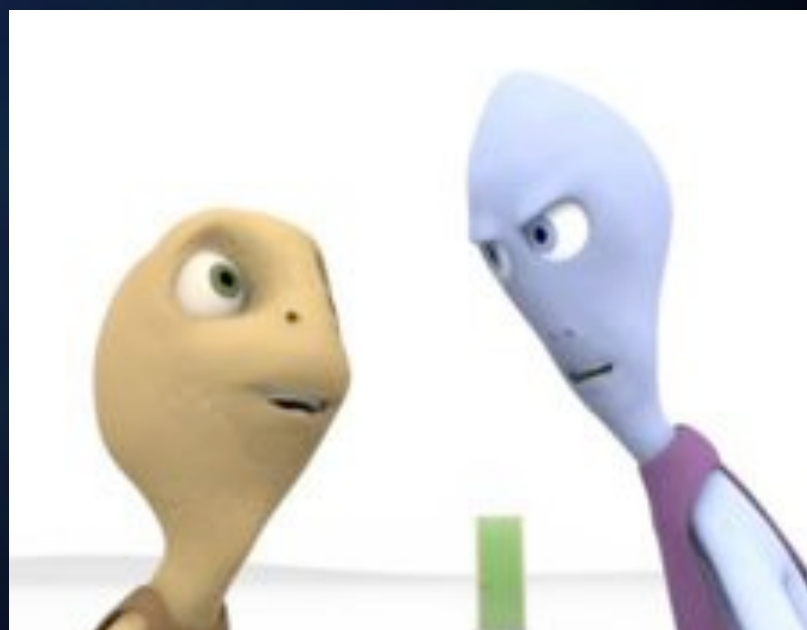
www.siggraph.org/publications/video-review/
ucbugg.berkeley.edu/

For each of these, feel free to shout "Amen, Brother!", and why it did / didn't work for you

I show SIGGRAPH animations before every lecture

I talk about opportunities for students to join my graphics group (UCBUGG)

The videos are available for \$40/video to ACM members



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

4/24



Team-teach (Meta)

gsi.berkeley.edu/resources/discussion/fiveways.html

“Classroom Observation:
The Observer as Collaborator”

–LuAnn Wilkerson

Rather than having a single lecturer / TA cover every lecture / section, if two can attend each other's sections, pair up & alternate weeks.

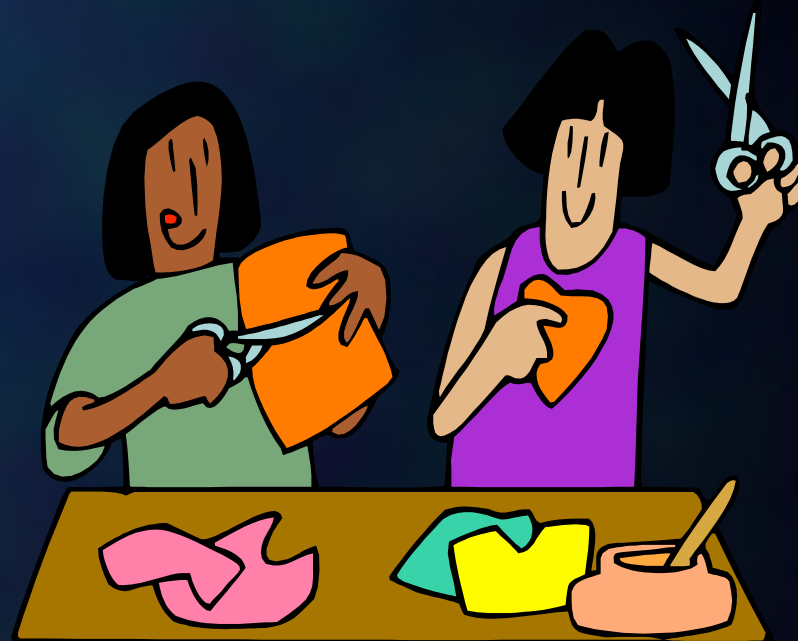
ON week they do everything (TAs would cover both secs)

OFF week they sit in the ON's first section, as an observer, taking notes of how well the ON TA did, and debrief after

TAs Buddy TA in weeks 2 through n-1 (not first and last)

Learn tips from **each other!**

Can avoid stuff you hate doing (that co-inst may enjoy)





Clickers & Peer Instruction (Lecture)

mazur-www.harvard.edu/research/detailspage.php?ed=1&rowid=8

www.interwritelearning.com/products/prs/

www.einstruction.com

We instituted them in our lower-div classes

Cost: ~\$40/clicker



Two pricing models, subscription or purchase

Midway through lecture, ask "concept test"



Students vote w/clicker

2 min peer instruction time with neighbor

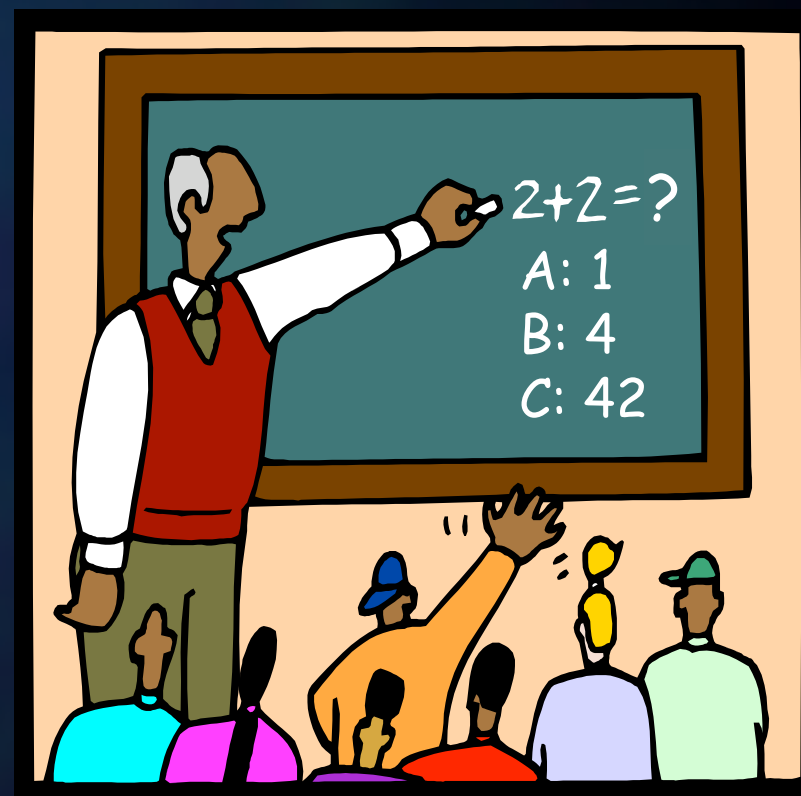
- Must reach consensus

"Team" votes again

We discuss answers

Real-time feedback!

Can ask real-time Q!





Empower TAs (Staff)

inst.eecs.berkeley.edu/~cs61c/
www.chezpanisse.com

Ask TAs teach a lecture

Have TAs and readers help write, beta-test, grade exams

Let TAs participate in syllabus discussions

Open your life to your staff

Share cell numbers (great for emergencies)

Grade exam at your home

Take your admins out to a fancy lunch every term





Enlightened Grading (Grading)

Absolute grading, bump up
at the end but never down
Allow later exams to replace
earlier exam grade(s)
Give EPA! sprinkle points

Effort

How much student tried
Office hours? Does all hw?

Participation

Does the person ask Qs in
lecture or discussion?

Altruism

Helping others in lab /
newsgroup / office hours

EPA grades are **hidden**, and
can boost up ~ 1/2 +/- grade





Bring Laptops to Exams (Exams)

developer.apple.com/documentation/Darwin/Reference/ManPages/man1/say.1.html

hogbaysoftware.com/product/clockwork

Visible timer

Interrupt vs polling!

Show exam bug
corrections on screen

...and can fix right there

Auto-announce time left

```
echo 'say "You may  
begin."' |  
/usr/bin/osascript  
sleep 1800 # 1/2 hour  
echo 'say "Sorry for  
the interruption... You  
have two hours left!"'  
| /usr/bin/osascript  
... etc ...  
plaympeg ./epilogue.mp3
```



Conclusion, pt I

I presented 6 “hidden”
tips from our collection

Lecturing

Office (hrs)

Staff

Exams

Labs

Section

Projects

Grading

Meta

As the Borg say:

Contribute your uniqueness
to the collective (our Wiki)



A Survey of Intro CS Curricula

www.cs.berkeley.edu/~ddgarcia/teaching/introsurvey

Goal: Understand what top US PhD-granting schools did in their intro CS curricula

Survey sent out, collected, results presented

Today I'll talk about the best practices we noticed



Jeff Forbes
Duke Univ



Dan Garcia
UC Berkeley



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

11/24



Schools (2006 "US News & WR")

- | | | | |
|-----|---------------------------|-----|-------------------------|
| 1. | CMU | 15. | U Michigan-Ann Arbor |
| | MIT | 18. | Columbia |
| | Stanford | | Purdue - West Lafayette |
| | UC Berkeley | | U Penn |
| 5. | Cornell | | Yale |
| | UI Urbana-Champaign | 22. | Brown |
| 7. | U Washington | | Rice |
| 8. | Princeton | | UNC - Chapel Hill |
| 9. | UT Austin | 25. | Duke |
| | U Wisconsin-Madison | | U Mass - Amherst |
| 11. | Cal Tech | | USC |
| | Georgia Tech | 28. | Johns Hopkins |
| 13. | UC San Diego | 29. | NYU |
| | U Maryland - College Park | | Rutgers - New Brunswick |
| 15. | Harvard | | UC Irvine |
| | UCLA | | Virginia |





What works really well and what is broken?

Brown

High TA-student ratio
Intro Experience Disparity

Cal

Closed required labs
Undergrad teaching staff
More pair programming

CalTech

CS11 is several minicourses.
Each student only takes 1 of these a term (of their choice)
A course can be repeated three times for credit

CMU

All courses taught in small sections – no large lectures!

Cornell

DrJava demos during lecture
Closed required labs
1/2 Matlab, 1/2 Java

Georgia Tech

(Compared to peers) very low attrition

90% success in Media CS2

No closed labs

Harvard

Teacher training

Princeton

Appl to commerce, science, engineering w/real datasets

Too much to cover!

Huge increase in enrollments





What works really well and what is broken?

Purdue

Lecture, lab

Hard to have creative projects
but consistent grading

Rice

Some want to move intro
Scheme course to Java

Easier later transition

But, wide background variety!

Rutgers

Online exams

Java as intro language!

Stanford

Undergrad section leaders

Staffing in computer cluster
while students work on
assignments, which are good

We (all of us) still tuning CS1 Java

UC Irvine

Informatics curriculum

UCLA

Getting students to program to
specs

Intro experience disparity

UCSD

Accelerated intro course track

Closed labs

Teaching memory allocation to
beginners

UIUC

All courses!

Undergrad-led reviews

Assembler EOS competition

Programming studio structure

Low-level understanding

New student communities





What works really well and what is broken?

U Penn

Compelling programming assignments

USC

Intro experience disparity

UT Austin

Students move well through CS2

Java doesn't allow them to learn memory management!

One course is "geeked out" which could turn off non-programmers

UW

Procedures-first intro to OOP

U Wisconsin

Information overload

Reduce sources of info

Active and collaborative learning!

"Information overload has been mentioned repeatedly to me by students as a principle problem in many of their courses.

It is not uncommon for courses today to have a textbook or two, course notes, as well as lecture slides which overwhelm students with presentations of the same or similar material.

Add to that course email, web pages/links, blogs, etc. and students get trapped spending too much time reading and reviewing and insufficient time actively applying their newly gained knowledge.

What has worked well for us is to reduce the sources of information and to direct students' efforts to more active and collaborative learning both in and out of class."





What are the innovative techniques being deployed?

www.eecs.mit.edu/images/EECS_VI_Fall06_FINAL.pdf

www-tech.mit.edu/V126/N63/63course6.html

MIT

Overhauling EECS curriculum

Had been unchanged for nearly 30 years

6.001 (Scheme) out, **Python in!**

6.01 and 6.02 to be new mandatory intro courses

Offer intro to dept

Cut broadly across EECS

Strong hands-on component

Build robots, apply concepts like radio, Fourier transform, networks, large systems

Then choose 3 or 4 subjects from foundation courses

E.g., Circuits, Signals, Systems





What are the innovative techniques being deployed?

www.cm.gatech.edu

Georgia Tech

Computational Media degree!

Different entry points to major

CS1 for different majors

Robotics

Media Computation

Flexibility

CS Majors can take any CS1

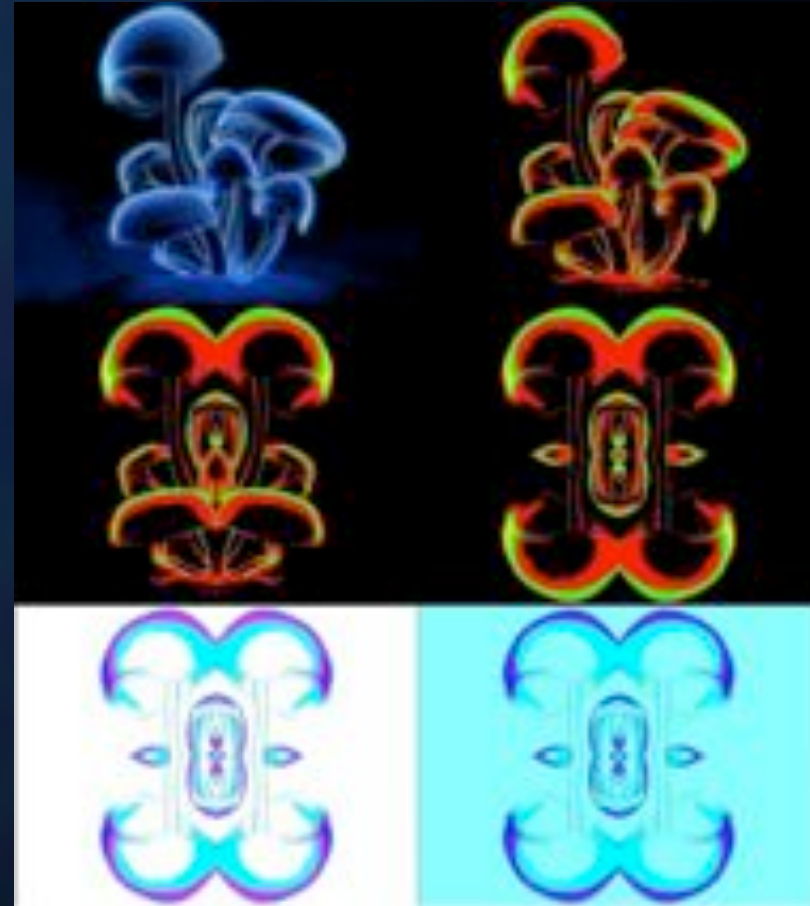
Contextualization

CS1s are typically dry!

Students today use flickr,
youtube, etc -- leverage that!

Allow the data to be THEIRS

movies, sounds, pix



Wil Justis, Milton HS



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

17/24



What are the innovative techniques being deployed?

inst.eecs.berkeley.edu/~selfpace

www.ucwise.org

UC Berkeley

UC-WISE

Web-based learning system
Interactive, collaborative & assessment

Decrease passive learning in lecture (Old: 3 Lec, 2 Lab)

Increase active learning in lab (New: 1 Lec, 6 Lab)

Top-down instruction

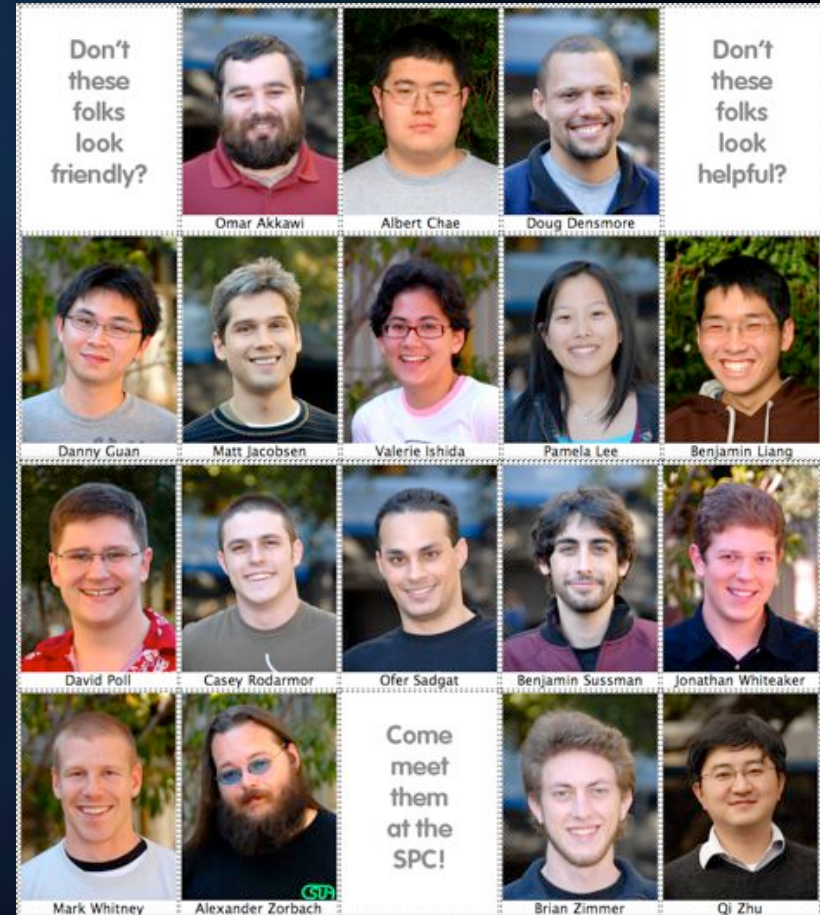
Scheme, Java, C, Assembler

Self-paced courses

1-unit course for programmers in Matlab, Pascal, C, Scheme, Unix, C++, Java, Python

Intro to programming in Scheme

Awesome undergrad tutors!



Conclusion, pt II

There is so much more analysis I didn't present!

It's all online; I encourage you to see our results

There is some great innovation in the US!

MIT, GaTech, UC Berkeley, Caltech, UIUC, ...

It would be great to learn about the innovation going on here to take back with me!



ACM Education Activities

acm.org/education

The ACM Education Board has three active task forces

Pipeline Crisis, Image

This was "enrollment"

Technology and Tools

Curriculum

As a member of the Board, I want to learn what your problems are and how ACM could help!



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

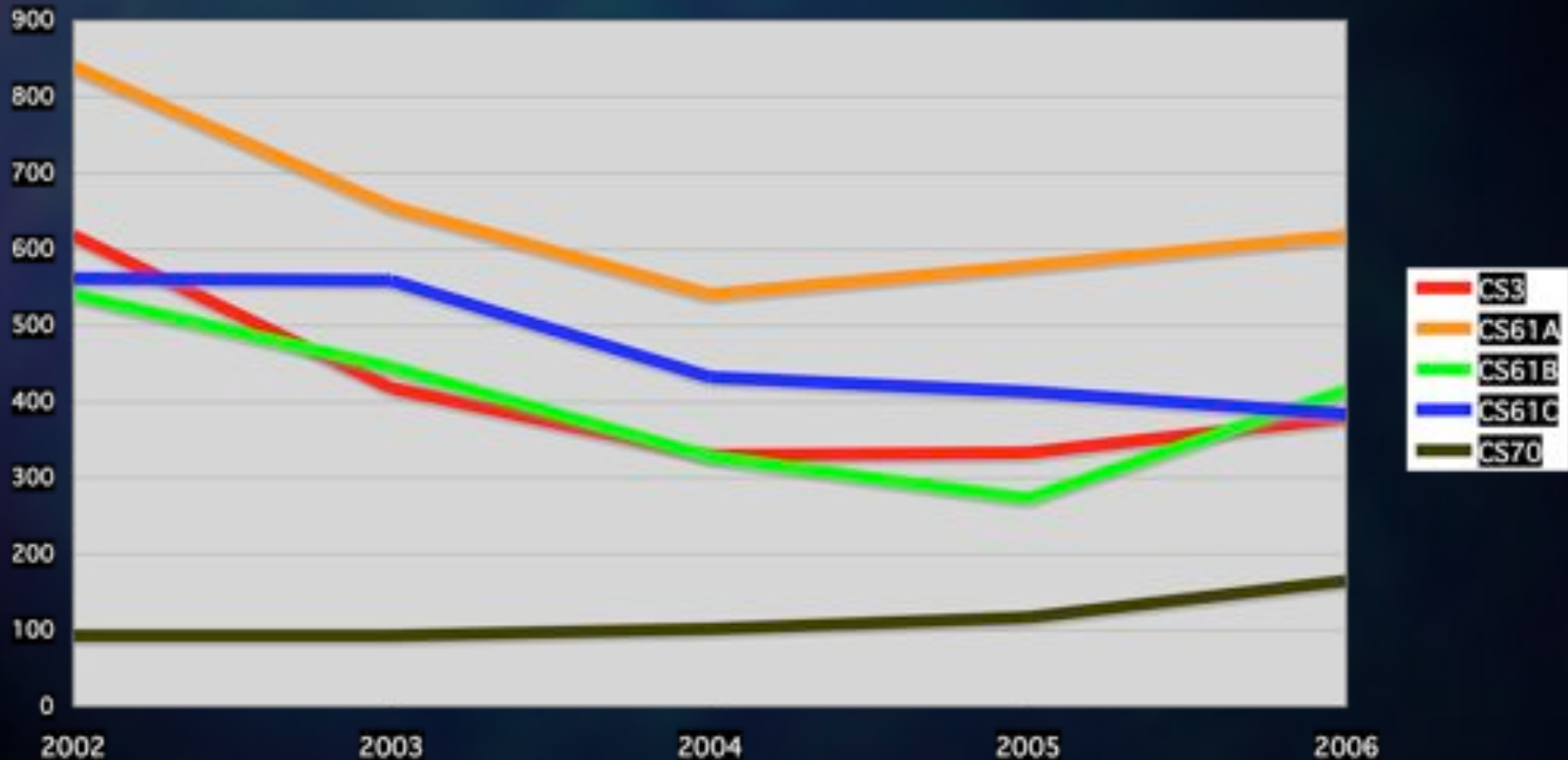
20/24



One Take on Enrollment

- Our numbers roughly half that of the 2000 peak

UC Berkeley Computer Science Annual Enrollment, 2002-present





Response : Image Site

computingcareers.acm.org

Numbers across US half that of 2000

Some coming back

Need for computing professionals growing

Computing image poor

Nerds, no life, just coding

Computing Degrees & Careers site launched

Supplemental brochure sent to all US high schools





Technology and Tools Site

Final Location: techtools.acm.org

Currently: www.cs.berkeley.edu/~ddgarcia/techtools

Problem

Computing educators (newbies and vets) don't know of all available classroom technology and tools (too hard to keep up)

Solution (still in beta)

Build site for them

Just links w/comments

What is missing?

Support Web 2.0

Ratings, tags, how it worked for you, etc.

ACM Home | Join | Renew | Subscribe

Association for Computing Machinery
Advancing Computing as a Science & Profession

Technology and Tools for Computing Educators

Welcome, computing educator! The goal of this site is to provide you with a vast array of ready-for-prime-time technology and tools resources to use in your class. We divide these into two categories below: those used in class (on-stage) and those used outside of class (off-stage). For the resources shown in *italics*, we present a discussion page with more details about the technology and an opportunity to share your experiences with other educators. Note that this is not an endorsement of any of the resources by ACM. We [welcome your feedback, updates or omissions](#).

On-stage resources	Off-stage resources
<ul style="list-style-type: none"> Basics <ul style="list-style-type: none"> Projectors, now with <i>wireless connections & screens</i> Presentation remotes from <i>Keyspan, Logitech</i> Laser pointers in <i>Red, Green, Bright Green, Green-Red and Blue</i> Presentation Software <ul style="list-style-type: none"> Microsoft's <i>Powerpoint</i> <i>OpenOffice.org</i> Apple's <i>Keynote</i> Google's <i>Presently</i> Tablets for Presentation <ul style="list-style-type: none"> <i>Ubiquitous Presenter (was Classroom Presenter), Livenotes</i> Classroom Response "Clickers" <ul style="list-style-type: none"> <i>elInstruction, Interwrite PRS, iClicker, TurningPoint, Quizdom</i> <i>One school did a 4-way shootout</i> <i>Often used with Peer Instruction and Just-in-Time Teaching</i> Program Visualization and Algorithm Animation <ul style="list-style-type: none"> <i>Java Program Visualization: Jelit</i> <i>Java-Hosted Algorithm Visualization Environment (JHAVE)</i> <i>The Complete Collection of Algorithm Animation</i> Sans-Computer Activities <ul style="list-style-type: none"> <i>CS Unplugged</i> <i>Kinesthetic Learning Activities</i> Understanding the Limitations of Technology <ul style="list-style-type: none"> <i>Jeffrey R. Young's When Good Technology Means Bad Teaching</i> <i>Edward Tufte's PowerPoint is Evil, Cognitive Style of Powerpoint</i> 	<ul style="list-style-type: none"> Course Management Systems <ul style="list-style-type: none"> <i>Blackboard & WebCT, Sakai, Moodle, others</i> <i>Edutools comparisons of current products</i> Grades and Submissions Management <ul style="list-style-type: none"> <i>Using spreadsheets, Excel for grades management</i> <i>Stephen H. Edwards' Web-CAT</i> Curriculum Repositories <ul style="list-style-type: none"> <i>Open Education Resources (OER)</i> <i>Creative Commons (CC)</i> <i>MIT's OpenCourseWare</i> <i>ArsDigita University's ADUni.org</i> <i>Rice University's Connexions</i> <i>UC Berkeley's UC-WISE</i> <i>Google's CS Curriculum Search, Tutorials & Sample Courses</i> <i>Intel's Technology Curriculum & K-12 Teaching Resources</i> Introductory Programming Instruction <ul style="list-style-type: none"> <i>Nick Parlante's JavaBat</i> <i>Turing's Craft CodeLab and its free service Chauncy</i> <i>ACM Java Task Force resources</i> Project Inspiration <ul style="list-style-type: none"> <i>Nifty Assignments</i> <i>Cut-the-Knot</i> <i>ACM's Programming Contest Problems</i> Utility Computing for Virtual Labs <ul style="list-style-type: none"> <i>Amazon's Elastic Compute Cloud (EC2)</i> <i>Google's Distributed System Tools</i> Video Lectures & Podcasts <ul style="list-style-type: none"> <i>Overviews: UCLA's What is Podcasting, Apple's Making a Podcast</i> <i>Google's Video Lectures</i> <i>UC Berkeley's Weiner Lecture Archives</i> <i>Peteris Krumin's Collection of Links</i> Anti-Plagiarism Software <ul style="list-style-type: none"> <i>Alex Aiken's MOSS</i> Other Technology and Tools Resources <ul style="list-style-type: none"> <i>UC Teaching, Learning and Technology Center (TLTC)</i> <i>4Teachers.org Teach With Technology</i>

ACM Education Council. Feedback? Updates? Omissions? Contact our [Webmaster](#)
Last Updated: 2007-10-11 @ 05:11 GMT



Teaching Tips, Best Practices, and Other Initiatives to Improve CS Education

www.cs.berkeley.edu/~ddgarcia

23/24

Conclusion, pt III

Take-away messages

Teaching tips

What are yours?

Contribute to our wiki



Best practices

What are yours?

Read survey for more



ACM initiatives

How could ACM help?

Talk to me about issues!



Let's collaborate!

