Getting started

Background

The course design project will be to design a ten-week course for somewhere in the computer science or electrical engineering curriculum. This assignment is a start on the project.

Readings

"Design Issues for Learning Environments", by Allan Collins. (available on Piazza as InternationalPerspectivesOnTheDesignOfTechnologySupportedLearningSystems.pdf).

"Evolution of the CS lower-division courses: 1983-1997", by Michael Clancy (available on Piazza as ld evolution.pdf).

Assignment

Working with a partner, choose the course you want to design. It should run for ten weeks and should fit into a quarter system curriculum as a three- or four-unit course.

- Describe the course prerequisites and design a short entrance exam to help verify that students have the prerequisite knowledge.
- Identify the course content by giving a brief description of what's in each of the ten weeks. (Don't forget to leave time for exams, holidays, and end-of-course review.)
- Identify one or two courses for which your course would be a prerequisite, describe what instructors of those courses should expect of your students, and design short entrance exams to help verify that students have learned it.
- Identify the three concepts or techniques in your course that you expect to be the most difficult for students to learn. Explain why these three things should cause students trouble.
- Situate your course along the various spectra discussed in the Collins article (memorization vs. thoughtfulness, whole tasks versus component skills, etc.) and indicate which if any of the teaching methods he describes at the end of the article will be appropriate for your course.
- Find URLs of a couple of articles in CS or Engineering education journals or conference proceedings that are relevant to your course. (The proceedings of the annual ACM SIGCSE—Special Interest Group in Computer Science Education—symposium are probably a good place to look first, and *IEEE Transactions on Education* is a good starting point for EE students.) For each article, provide a brief summary, evaluation, and description of how it will enlighten your course design.

Put all this on the Google Docs homework for our class by Saturday, February 1. Be prepared to review and comment (using Google's "Insert→Comment" feature) on the documents of other participants on the next two days. We'll build on those comments when we meet on Monday, February 3.

Misconceptions and interfering attitudes

Reading

"Misconceptions and Attitudes that Interfere with Learning to Program", by Michael Clancy, in *Computer Science Education Research*, Marian Petre and Sally Fincher (editors), London: Routledge Falmer, 2004.

If you enjoyed that reading, you might also enjoy… (optional)

Both these articles are described briefly in "Misconceptions and Attitudes ...".

- "Mind Your P's and Q's: Using Parentheses and Quotes in LISP", by Elizabeth Davis *et al.*, *Empirical Studies of Programmers: Fifth Workshop*, Cook *et al.* (editors), Ablex, 1993 (not accessible online).
- "Student Beliefs about Pascal Programming", by Ann Fleury, *Journal of Educational Computing Research*, volume 9, number 3, 1993, pp. 355–372 (not available online).

Assignment due February 8th

Davis *et al.* describe *misconceptions* that students learning about parentheses and quotes in a Lisp course develop. The misconceptions might arise, say, from inappropriate generalization from examples or from inappropriate transfer from other knowledge areas.

- List some misconceptions that students might develop while trying to learn one of the difficult concepts or techniques you described in assignment 1. Also provide a set of exercises or class activities that would expose these misconceptions.

Fleury describes student *attitudes* that can interfere with their learning.

- Suggest some ways in which attitudes students have developed, either from earlier courses or from earlier experience outside school, might interfere with their learning the material in your course.

Assignment due before class on February 10th

Comment on the misconceptions, attitudes, and corresponding exercises and activities that the other 302 participants have devised.

Concept maps

Reading

"A Primer on Concept Maps", by Robert K. Noyd, *The Journal of Cooperation and Collaboration in College Teaching*, volume 10, number 1 (winter 2000); available on the web as http://academic.wsc.edu/frc/innovations.htm. This article is intended to give you some idea of what concept maps are and how one might use them in education. Googling "concept map" will reveal a bunch of other resources.

We've created a sample concept map to explain "interrupt handling" (say, in CS61C); you can find it on Piazza under *resources* entitled: InterruptHandlingConceptMap.pdf

Assignment due February 15th

Create a concept map PDF for one of the hard concepts and misconceptions you described in the first two assignments. (Do not create a concept map for your *entire class*; focusing on one smaller "nugget" will allow you to focus on this again in a later assignment.) You are not required to use a particular graph visualization/editing tool, but may find the excellent GraphViz software (http://www.graphviz.org/) helpful. Upload the PDF to your Google Docs folder and link to it from the *Homework* Google Doc.

Assignment due before class on February 13

Examine the concept maps of the other CS 302 participants and suggest more nodes or links, or different ways to organize them. Make these suggestions in the *Homework* Google Doc since the Google Apps don't currently allow "marking up" PDF files.

Course exercises

Reading

Tools for Teaching, "Homework: Problem Sets", chapter 27 (1st ed), 37 (2nd ed).

Optional Reading

"Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving", Katerine Bielaczyc, Peter L. Pirolli, and Ann L. Brown, Cognition and Instruction (1995), volume 13, number 2, pages 221–252. Available online as http://www.jstor.org/stable/3233714.

Assignment due February 22nd (part 1) and March 1st (part 2)

Part 1 (due February 22): In *Taxonomy of Educational Objectives, Handbook I: Cognitive Domain* (McKay Publishing, 1956), B.S. Bloom proposed a hierarchy of categories of questions or exercises. Anderson and Krathwohl, in *A Taxonomy for Learning, Teaching, and Assessing* (Longman, 2001) refined this into a two-dimensional grid, with the "knowledge dimension" on one axis and the "cognitive process" dimension on the other. The grid is shown below. Various Web sites (e.g. http://www.uwsp.edu/education/lwilson/curric/newtaxonomy.htm) provide more details about the meaning of the terms, but don't obsess over the definitions.

	The cognitive process dimension					
The knowledge dimension	Remember	Understand	Apply	Analyze	Evaluate	Create
Factual knowledge						
Conceptual knowledge						
Procedural knowledge						
Metacognitive knowledge						

By the end of the first week, provide at least one exercise in each row and each column of the grid positions (you could have full coverage with only six exercises). Try to spread them throughout your course, e.g. by having one exercise from each category in each two-week period. Students in previous semesters have found this difficult, so don't be surprised if you do too.

Part 2 (due March 1): An activity not really covered in the table above is *reflection*, learning from experience. The paper by Bielaczyc *et al.* describes some of the benefits of a particular kind of reflection, that of *explaining* what one has learned. Provide more exercises in any of the remaining empty grid positions (so that every row and column contain *two* exercises) and include some opportunities for reflection in your list of exercises.

Assignment due before class on February 24th

Review each other's exercises and provide some constructive criticism.

Case studies

Readings (in ranked priority of what we suggest you read)

"The Case for Case Studies of Programming Problems", by Marcia Linn and Michael Clancy, *Communications of the ACM*, volume 35, number 3, pages 121-132, March 1992 (accessible from the ACM Digital Library). http://doi.acm.org/10.1145/131295.131301

"Case Studies in the Classroom", by Michael Clancy and Marcia Linn, revised version of a paper appearing in the proceedings of the 23rd SIGCSE Technical Symposium on Computer Science Education, Kansas City, Missouri, March, 1992; published as *SIGCSE Bulletin*, volume 24, number 1, March 1992 (accessible from the ACM Digital Library).
http://doi.acm.org/10.1145/134510.134554

"How Experts Differ from Novices", chapter 2 in *How People Learn: Brain, Mind, Experience, and School* (expanded edition, 2000). Available online as
http://www.nap.edu/openbook.php?record_id=9853&page=31.

Tools for Teaching, chapter 19 (1st ed) and 24 (2nd ed).

Sample case studies

"Difference Between Dates", by Michael Clancy and Marcia Linn (accessible via Piazza as Scheme+DBD.pdf).

"Linking Shoes Together" and "A Sort of Debugging", excerpted from *Designing Pascal Solutions: Case Studies with Data Structures*, by Michael Clancy and Marcia C. Linn, W.H. Freeman and Company, 1996 (accessible via bspace as 6. pointers.pdf and 7. list debugging.pdf).

Assignment due March 8

Choose one of the homework exercises you've designed already, and produce an outline of a case study of its solution. Attempt to identify all applications of expertise in the solution and why the expertise was applicable at each point. (The excerpt from *How People Learn* may suggest some ideas.) Also identify decision points one might encounter in the solution and criteria an expert might use to make the decisions.

Assignment due before class on March 10

Attempt to locate places in one of your fellow students' case studies where the author failed to identify some application of expertise or failed to justify some decision made in the solution.

Collaborative learning activities

Readings

Theoretical basis

"Pedagogies of Engagement: Classroom-Based Practices", Karl A. Smith *et al.*, *Journal of Engineering Education*, January 2005 (http://onlinelibrary.wiley.com/doi/10.1002/j.2168-9830.2005.tb00831.x/abstract);

Pedagogical strategies (in ranked order of how we suggest you go through them)

"Effective Strategies for Cooperative Learning", Richard M. Felder and Rebecca Brent, *Journal for Cooperation and Collaboration in College Teaching*, volume 10, number 2, spring 2001 (http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/CLStrategies(JCCCT).pdf);

Tools for Teaching, chapter 18 (1st ed) and chapters 21-23 (2nd ed).

"Experiments with Industry's 'Pair-Programming' Model in the Computer Science Classroom", by Laurie A. Williams and Robert R. Kessler (*Computer Science Education*, 2001, volume 11, number 1, pages 7–20) (http://dx.doi.org/10.1076/csed.11.1.7.3846).

"A Design for Team Peer Code Review", by Deborah A. Trytten (*SIGCSE '05 Proceedings,* pages 455–459; accessible through ACM Digital Library). (http://doi.acm.org/10.1145/1047344.1047492).

"The Nuts and Bolts of Cooperative Learning in Engineering", Nikos J. Mourtos, Proceedings of the 1994 IEEE Frontiers in Education Conference, pages 624–627 (http://ieeexplore.ieee.org/iel2/4449/12614/00580621.pdf?arnumber=580621).

Assignment due March 15

Identify activities among those you have designed already that would be appropriate for doing in a group, and explain why students would learn more from doing them with partners than by themselves.

Design one or two activities (e.g., switching roles during pair programming) to improve group performance (e.g., to remedy a group dynamic problem, or to increase efficiency and fairness).

Design an exam problem that would be appropriate for group solution, and justify your design.

Before class on March 17

Review each other's activities and comment on them.

Grading policy

Readings

Tools for Teaching, chapters 23, 32, and 33 (1st ed), or 32, 43, and 44 (2nd ed)

Assignment due March 29th

Design and justify a grading policy for your course. State your overall goal – what are you using grades to do? In particular, defend the following:

- your choice between grading on a curve and grading on an absolute standard;
- the relative weights you assign to homework, labs (if any), exams, and other activities;
- your use of extra credit assignments or the "best M out of N scores", or a "clobber" policy wherein a higher grade on a later comprehensive exam can replace an earlier score;
- your evaluation of assignments or exams done in partnership.
- other issues. E.g., your policy for someone who misses an exam, your policy for cheating cases (which may differ for labs, homework, projects, and exams), etc.

Explain how you would decide a course grade for a student near a borderline in your grading scale, e.g. the best of the B+ students. If you would decide a borderline grade differently for different borderlines or for different distributions of scores comprising the final score total (say, someone doing very well on exams with a low homework score versus someone with the opposite distribution), explain how you would do so and defend your policy.

Before class on March 31st …

Review each other's grading policies, looking for advantages and disadvantages that the policy's designer may have overlooked.

Exams

Reading

Tools for Teaching, chapters 28-31 (1st ed), or chapters 39-42 (2nd ed).

Assignment due April 5th

Design the first course exam. Try to include questions from each of Bloom's categories (recall; comprehension, translation, interpretation; application; analysis; synthesis; and evaluation). Include a problem based on your case study if it is topically appropriate. Also list the goals for each question, and provide intended solutions. Finally, estimate how long each question will take your students to complete.

Before class on April 7th …

Review each other's exams, looking for places where questions are not clear or are likely to take longer than estimated to answer.

Online Education (MOOCs)

Readings

"Report on the CCC-CRA Workshop on Multidisciplinary Research for Online Education", (Report prepared by Douglas H. Fisher and Armando Fox), CRA, February 12-13, 2013. Available at http://www.cra.org/ccc/files/docs/CCC-MROE-Report.pdf

"xMOOC vs cMOOC", by DegreeofFreedom on April 29, 2013 in *Online Learning*. Available at http://degreeoffreedom.org/xmooc-vs-cmooc/

Assignment due April 12th

What would it take to put your course online, as a MOOC? This might include interactive & adaptive online assessment, Web 2.0 constructs (e.g., tagging, rating, commenting, remixing), learning community formation and productive student-student interactions, building systems that scale to 100k students, etc.

While you are considering creating your own resources, consider what resources already exist. Are there current *textbooks* whose readings or materials you could leverage? Are there other *online* resources (demos, tutorials, data sets, Wikipedia, media, etc.) you might consider?

Post your ideas Saturday in time to give us all a chance to comment on them by class on Monday.

Before class on April 14th…

Review each other's activities.

Reflection

Reading

Tools for Teaching, chapter 58.

"Evaluating and Documenting Teaching Effectiveness" from *Teaching at Its Best* (second edition), by Linda B. Nilson, Anker Publishing, 2010 (available on Google Books online in snippets).

Assignment due April 23rd

1. Collaboratively (with each other) design your own CS 375. Produce a short consensus Google doc that identifies what is done in each session and why. The course you design should be a 2-unit course.

   The activities in recent offerings of CS 375 are listed at the web site inst.eecs.berkeley.edu/~cs375. The campus guidelines for GSI training and mentoring along with other material useful for a 375 course are at http://gsi.berkeley.edu/faculty/.

2. Write a statement of your teaching philosophy. Suggestions for what it should include are at http://gsi.berkeley.edu/faculty/facwithgsis/idea_11.html and in this week's readings. Include it on your CS 302 web site.

3. Please help us improve CS 302 by answering the questions below. As with your CS 375 design, this should be a **collaborative** activity. Post the answers on the wiki.

   - Which assignments were especially good? Which were the worst? Please explain.
   - Which readings were especially good? Which were the worst? Please explain.
   - Which class discussions were especially good? Which were the worst? Please explain.
   - Should class discussions have focused more on homework or on the readings? What other activities would have provided a good use of class time?
   - What should we change the next time we offer CS 302?

   Comments on any other aspects of the course would also be helpful.

4. Finally, send us mail (to clancy@cs, ddgarcia@cs, and ayazifar@eecs) answering two final questions:

   - To what extent have your expectations about and attitude toward teaching changed as a result of CS 302? Please explain.
   - Estimate how much work would remain for you to teach the course you designed. Should an assignment have addressed this?