

Minutes of the CSS Committee
January 11, 2013
10:30am-12noon, CP 206C

Attendance: Donald Chinn (co-chair), John Mayer (co-chair), Matt Alden, Zaide Chavez, Ling Ding, Alan Fowler, Robert Friedman, Bryan Goda, Beth Jeffrey, George Mobus, David Ross.

1. The committee approved the minutes from Dec. 7 meeting.
2. **CSS curriculum reform discussions.**

The committee discussed issues surrounding the proposed framework for the new curriculum that was distributed prior to the meeting. (See attachment to these minutes.) After some discussion, it was generally agreed that an inner/outer core structure to the curriculum would be an improvement over the current structure. The main advantage of the inner/outer core structure is to allow students more flexibility in choosing what parts of computer science they wish to learn about. Logistically, the new structure will allow for some more flexibility in scheduling classes.

Mobus suggested two possible changes to the proposed structure:

1. In the systems/hardware part of the inner core, he suggested that perhaps 333 should be the prerequisite for 371 (rather than the other way around as currently proposed) so that knowledge of C could be leveraged in later courses, including 371.
2. He suggested that since mastery of programming is fundamental to the rest of the curriculum, 305 could be a prerequisite to both 371 and 321.

Some other issues that were discussed:

- 400-level electives: should we allow CSS students to take ITS 400-level electives and count them as CSS 400-level electives for graduation purposes? There was some concern expressed that when CSS students take ITS courses, the class has two different “cultures,” which affects the teaching of the class.
- Since outer core courses can count toward 400-level electives, perhaps they should all just be called “electives.”
- Similarly, should we rename the outer core courses to all be 300- or 400-level?
- Jeffrey pointed out that with all the new choices students will need to make, we will need to create materials to help students make choices, such as planning grids and a document that suggests sequencing of courses for students.
- There was general agreement that use of the Moodle discussion board would facilitate discussion of the various issues before the next meeting. Chinn will create threads on Moodle to discuss different issues.

3. Admissions. Friedman is working on setting an early priority deadline for admissions. Admissions and Enrollment is going through some major organizational changes. When things are more stable there, he will pursue the idea with them. Jeffrey and others also expressed concerns about the current admissions process and its impact on scheduling and student success.

Ideas for the New CSS Curriculum

DRAFT: January 5, 2013

Motivation for Change

Several changes in the composition and growth of the different programs within the Institute have prompted the need to reexamine and reorganize the CSS undergraduate curriculum. When the current CSS curriculum was put in place (Autumn 2005), there was only one undergraduate program (CSS) in the Institute. Now there are three undergraduate programs (CSS, CES, ITS). Students who would have majored in CSS when CES and ITS were not options are now enrolled as students in those majors. The systems courses in the CSS program (TCSS 371, 372, 422) have evolved recently to serve the needs of CES majors, possibly to the detriment of both CES and CSS students. As a result, the CES program has decided to have a separate introductory programming sequence and systems sequence.

The current CSS undergraduate curriculum has a large number (10) of courses in the core, leaving little room for students to tailor their education to how they intend to apply their knowledge and skills after graduation. We want students to have a broad education, but we also want them to go into depth in areas of interest to them. We want expose our students to core computer science principles while simultaneously giving them the opportunity to acquire skills that will be instrumental in finding jobs in the computer technology industry.

A Proposal for the Organization of a New Curriculum

What follows below is a proposed organization of courses for the B.S. degree in CSS.

Introductory programming sequence (TCSS 142, 143). The scope and purpose of these courses will essentially remain the same. They offer transfer students a path to the CSS major that can be tailored to what programming experience they have received at their community colleges. It also offers non-transfer students a direct path to the CSS major.

Inner core (TCSS 371, 333; TCSS 305, 342; TCSS 321; TCSS 325). These are required courses (i.e., all CSS students must take all of them). The courses represent fundamental concepts in computer science (machine architecture; programming practicum; data structures; discrete mathematics, and computers, ethics, and society). The purpose of the new course, TCSS 333 (systems programming), is to introduce students to C programming in the context of systems programming. The course will be roughly modeled after UW Seattle's course of the same name, but tailored to fit with our curriculum. [30 credits]

Outer core. These are courses that no one of which is required, but that every student must take some of. The courses are roughly grouped into the three "traditional" areas of computer science, hardware, software, and theory. Each CSS student would be required to take at least one course from each of the three areas. [15-30 credits]

Hardware/systems: TCSS 372 (computer architecture), TCSS 422 (operating systems, prerequisite TCSS 333 and 342).

Software: TCSS 360 (software development and quality assurance), TCSS 445 (databases)
Theory: TCSS 343 (design and analysis of algorithms, prerequisite TCSS 342), TCSS 440 (formal models in computer science, prerequisite TCSS 342)

TCSS 445 doesn't quite fit into the "software" category, but please keep in mind that these categories are loose. Note that TCSS 322 (discrete mathematics II) would be eliminated from the core and the models of computation parts of that course would be expanded into a full course on the theory of computation (TCSS 440).

400-level CSS electives. None of these are required, but the balance of the 75 Institute credits need to be chosen from this group of classes. [15-30 credits]

Any course in the CSS outer core can count towards the 75 unit total.

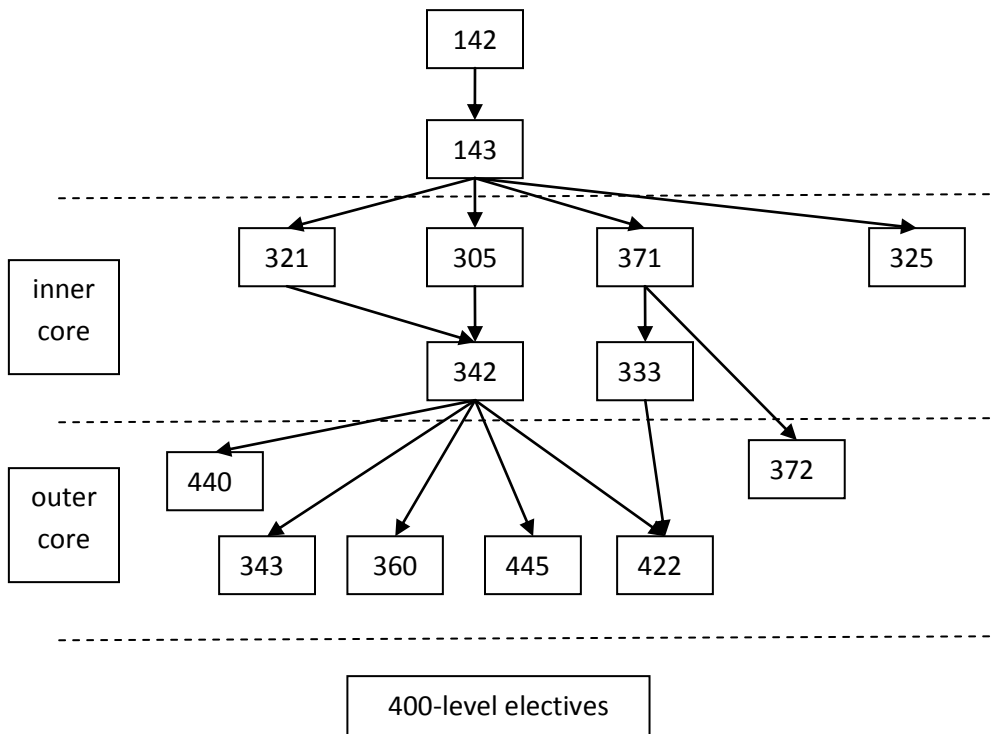
To graduate, a student must take at least 10 units of 400-level CSS lecture classes (not TCSS 497, 498, 499).

Up to 5 units from 400-level non-CSS Institute (CES, ITS, TINST) classes may be taken as a substitute for 5 units within this category.

Up to 10 units of TCSS 497, 498, and 499 may be taken to satisfy graduation requirements.

300- or 400-level non-Institute electives. To graduate, a student must take at least 15 units of 300- or 400-level non-Institute classes. [15 units]

The prerequisite structure for individual courses would not change (except where noted).



Areas of Flexibility

CS 0: Should this be a requirement for non-transfer students before taking TCSS 142?

TCSS 142: Should we teach it in Python or leave it (Java)?

TCSS 342: Is there an advantage (or a disadvantage) to alter the way it is taught (e.g., not language-specific)?

Graduation Requirements

(As articulated above.)

Logistics

- Equivalence between TCSS 142/143/305 and TCES 201/202/203. Proposal: Because the designs of the two programming sequences are so different, it is difficult (impossible) for a student to skip between the two sequences (e.g., take TCES 201, then TCSS 143 and 305). However, any student who has completed the TCSS or TCES sequence should be considered ready for TCSS 342 and satisfy the introductory programming sequence of either program.
- Pathways for CES students into TCSS 342. Proposal: the CES program should offer a short course (2 units) to introduce students to Java so that they do not have language issues in TCSS 342.
- Advising, suggested paths.
- Admissions, placement exams.
- How many times to offer each of the courses.
- Renumber all inner and outer courses to be 300-level?

Work Items

(Possible) Reworking or development of syllabi for the following courses:

TCSS 142

TCSS 143

TCSS 342

TCSS 333

TCSS 371

TCSS 372

TCSS 422

Outstanding Issues

- Is there a common course or sequence that all Institute students take before specializing in their respective majors?
- In particular, is there a common programming sequence that all Institute students can take (or at least CSS and ITS)?
- What should the requirements for the B.A. in CSS be?
- What should the requirements for the CSS minor be?