

II. A. CURRENT CATALOGUE COPY:

COMPUTER SCIENCE MAJOR, BA or BS DEGREE

A. General Education Core Requirements/38 credits.

Computer Science majors are required to take CMSC 350 for general education goal 12.

B. BA Degree Additional Degree Requirements/6 credits.

BS Degree Additional Degree Requirements/7 credits.

Students must take an additional 4 credit natural science course and either MATH 171 or MATH 270 to earn a B.S. degree.

C. Major Requirements/54-56 credits (plus 4 credits included in General Education)

Computer Science Majors must earn a C or better in any CMSC course for it to count towards the major.

Computer Science Foundations/23-25 credits (plus 1 credit included in General Education)

CMSC 160 Introduction to Algorithmic Design I/4 credits

CMSC 162 Introduction to Algorithmic Design II/4 credits

CMSC 301 Computer Organization/3 credits

CMSC 308 Organization of Programming Languages/3 credits

CMSC 310 Introduction to Operating Systems and Computer Architecture/3 credits

CMSC 461 Seminar in Computer Science/ 3 credits

ENGL 319 Technical Writing/3 credits

CMSC 490 or CMSC 492 or CMSC 498/1-3 credits (1 credit satisfies General Education Goal 14)

Mathematics Foundations/4 hours (plus 3 credits included in General Education):

MATH 261 The Differential and Integral Calculus I/4 credits

(3 of these 4 credits satisfy General Education Goal 5)

CMSC 300 Discrete Mathematics for Computer Science/3 credits

OR MATH 300 A Transition to Advanced Mathematics/3 credits

Electives*

Students must complete a total of 27 additional hours from 300-400 level Computer Science courses, and in so doing must complete one of the following concentrations:

Theoretical Concentration

CMSC 342 Systems Programming/3 credits

CMSC 415 Theory of Computation/3 credits

CMSC 420 (MATH 420) Graph Theory/3 credits

Choose one of the following:

CMSC 210 Web Page Design and Scripting/3 credits

CMSC 215 Introduction to Fortran/3 credits

CMSC 220 Advanced Java Programming/3 credits

CMSC 225 Introduction to Ada/3 credits

CIMS 372 COBOL Programming/3 credits

Choose two of the following:

CMSC 360 Computer Network Theory/3 credits

CMSC 362 Theory of Databases/3 credits
CMSC 445 Compiler Design/3 credits

Information Systems Concentration

CMSC 210 Web Page Design and Scripting/3 credits
CMSC 340 System Administration/3 credits
CMSC 355 Introduction to Computer and Network Security/3 credits
CMSC 362 Theory of Databases/3 credits
CMSC 408 Software Engineering/3 credits
CMSC 360 Computer Network Theory/3 credits
OR CMSC 445 Compiler Design/3 credits

*Electives may NOT include CMSC 350.

D. General Electives, BS degree/19-21 credits
General Electives, BA degree/20-22 credits

E. Total Credits Required for BA or BS degree/120

II. B. PROPOSED REVISED CATALOGUE COPY:

COMPUTER SCIENCE MAJOR, BA or BS DEGREE

A. General Education Core Requirements/38 credits.

Computer Science majors are required to take CMSC 350 for general education goal 12.

B. BA Degree Additional Degree Requirements/6 credits.

BS Degree Additional Degree Requirements/7 credits.

Students must take an additional 4 credit natural science course and either MATH 171 or MATH 270 to earn a B.S. degree.

C. Major Requirements/44-46 credits (plus 4 credits included in General Education)

Computer Science Majors must earn a C- or better in any CMSC course for it to count towards the major.

Computer Science Foundations/29-31 credits (plus 1 credit included in General Education)

CMSC 160 Introduction to Algorithmic Design I/4 credits

CMSC 162 Introduction to Algorithmic Design II/4 credits

CMSC 201 Computer Organization/3 credits

CMSC 208 Grammars, Languages, and Automata/3 credits

CMSC 242 Introduction to network and systems programming/3 credits

CMSC 262 Data structures and algorithms in application/3 credits

~~CMSC 308 Organization of Programming Languages/3 credits~~

~~CMSC 310 Introduction to Operating Systems and Computer Architecture/3 credits~~

CMSC 461 *Senior capstone in computer science*/3 credits

ENGL 319 Technical Writing/3 credits

CMSC 490 or CMSC 492 or CMSC 498/1-3 credits (1 credit satisfies General Education Goal 14)

Mathematics Foundations/3 hours (plus 3 credits included in General Education):

MATH 261 The Differential and Integral Calculus I/4 credits

(3 of these 4 credits satisfy General Education Goal 5)

MATH 175 Discrete Mathematics/2 credits

~~CMSC 300 Discrete Mathematics for Computer Science/3 credits~~

~~OR MATH 300 A Transition to Advanced Mathematics/3 credits~~

Electives*

Students must complete a total of 12 additional hours from 300-400 level Computer Science courses, and in so doing must complete one of the following concentrations: of Computer Science courses, including the following:

Choose one of the following:

CMSC 210 Web Page Design and Scripting/3 credits

CMSC 280 Programming in a Second Language/3 credits

Choose two of the following:

CMSC 355 Introduction to Computer and Network Security/3 credits

CMSC 362 Theory of Databases/3 credits

CMSC 415 Theory of Computation/3 credits

CMSC 420 (MATH 420) Graph Theory/3 credits

~~Theoretical Concentration~~

- ~~CMSC 342 — Systems Programming/3 credits~~
- ~~CMSC 415 — Theory of Computation/3 credits~~
- ~~CMSC 420 (MATH 420) — Graph Theory/3 credits~~

~~Choose one of the following:~~

- ~~CMSC 210 — Web Page Design and Scripting/3 credits~~
- ~~CMSC 215 — Introduction to Fortran/3 credits~~
- ~~CMSC 220 — Advanced Java Programming/3 credits~~
- ~~CMSC 225 — Introduction to Ada/3 credits~~
- ~~CIMS 372 — COBOL Programming/3 credits~~

~~Choose two of the following:~~

- ~~CMSC 360 — Computer Network Theory/3 credits~~
- ~~CMSC 362 — Theory of Databases/3 credits~~
- ~~CMSC 445 — Compiler Design/3 credits~~

~~Information Systems Concentration~~

- ~~CMSC 210 — Web Page Design and Scripting/3 credits~~
- ~~CMSC 340 — System Administration/3 credits~~
- ~~CMSC 355 — Introduction to Computer and Network Security/3 credits~~
- ~~CMSC 362 — Theory of Databases/3 credits~~
- ~~CMSC 408 — Software Engineering/3 credits~~
- ~~CMSC 360 — Computer Network Theory/3 credits~~
- ~~OR CMSC 445 Compiler Design/3 credits~~

*Electives may NOT include **CMSC 121** or CMSC 350.

D. General Electives, BS degree/29-31 credits
General Electives, BA degree/30-32 credits

E. Total Credits Required for BA or BS degree/120

III. RATIONALE.

General background. The computer science (CS) major was originally designed in the mid-1980s based on then-current guidelines issued by our professional societies (ACM and IEEE) in 1978. It has been tweaked several times since then but has been long overdue for a significant overhaul. The largest problems with it were that it had become too large (in number of required courses), had remained too flat (in terms of prerequisites), and that it was insufficiently flexible for students who diverged in any way from a "standard" 4-year plan (e.g. by transferring in, by studying abroad, or just by having a single really bad semester). Our goal in this program revision was to address these structural problems and also to update the courses themselves to reflect the needs of 21st century CS students.

Size of major. The current major is among the largest in CCAS, requiring 17 courses in CS alone, plus two specific math courses, a specific English course, and all the usual general education courses; many CS students only have room for three or four truly elective courses, which is a disservice to our liberal arts goals. It also makes it exceedingly difficult for our students to take on a minor or a double major, and with even the slightest scheduling mistake it forces our students into a ninth or tenth semester.

A big part of the reason the major had gotten so large was the desire to be sure our majors were exposed to the breadth of the field. This objective remains as important as ever, but where it has previously been achieved by many single-topic upper-level classes, it can be served just as well (or perhaps better) by a smaller number of breadth-oriented mid-level classes. This is seen particularly in the new courses CMSC 208, which will be a sophomore-level course combining content from the current 308 (Programming languages, major requirement), 415 (Theory of computation, concentration requirement), and 445 (Compiler design, concentration requirement); and CMSC 262, a sophomore-level course combining content from the current 362 (Databases, concentration requirement) and 389 (Artificial Intelligence, elective) as well as some areas not currently covered. By consolidating material from multiple areas into a smaller number of courses, we can safely reduce the number of courses required while producing graduates that are more well-rounded than before (both within CS and in a more general sense).

Flatness of prerequisites. The second big problem with the current major is that it is a little too "flat" in its prerequisite structure: after CMSC 162 (the second course in the major), nearly every other course becomes available. This creates a problem in that less-prepared sophomores are in the same classrooms as better-prepared seniors, for every course, which has the dual effect of burning out our sophomores and preventing even our seniors from achieving real depth in anything. It also causes significant variability in the same course from year to year, depending on the proportion of upper- or lower-level students that happen to take it that year.

On the other hand, it is also important that the prerequisite structure not become too deep: the longer the chain, the harder it becomes to schedule the most upper-level courses (and the more likely that a student would need to take longer than four years). Our design goal here was that a student transferring in with only the intro courses (CMSC 160 and 162) would be able to straightforwardly finish in two years (something that is not currently true, despite our too-flat prerequisite structure, due to some course rotation problems---so this even represents an improvement).

The solution here is the construction of a standard 200-level core. It includes the two new courses mentioned above (208, 262), as well as adapting the current 342

(Systems programming) into 242, a course that more explicitly prepares for 310 (Operating systems) and 360 (Network theory). To match the other courses in this group, we also renumber 301 to 201 (but otherwise leave it largely unchanged). We thus make structural use of the course numbering: upper-level courses (300 or 400) may require one (or occasionally two) of the 200-level courses as a prerequisite without worrying that this will create scheduling problems, but by adding the 200-level prerequisite we can guarantee that the students in the upper-level courses are more experienced and better prepared to explore the content more deeply.

This has a particularly nice effect in the case of the new courses (208, 262): since they include material that was formerly covered in the early weeks of other courses (308, 415, etc), those courses are now able to "hit the ground running" and get through much more material more effectively.

Flexibility. The last big problem with the current program is its lack of flexibility. Because of the large number of required courses and the small number of faculty to teach them, all students effectively need to take nearly every course the department offers. Some of those courses are offered only in alternate years; this makes it difficult to study abroad or take a leave of absence, and one bad term can be disastrous, delaying graduation by semesters or years.

Our solution to this problem was to reduce the number of specifically required courses (there are more in the "foundations" list, but no more concentrations), and then guarantee that those courses are taught every year. Together with the improved prerequisite structure, this means that a student can be off campus any particular semester without delaying their graduation, and since most of the required courses will be normally taken by the sophomore year, it should usually be possible to retake a poorly-understood course.

It also introduces flexibility for the faculty. Although during the transition to this new curriculum we will need to maintain our existing upper-level course list (until students on current catalogues graduate), there is no longer a structural requirement that we offer a specific list of courses. This can free up staffing slots to develop new upper-level courses to match faculty expertise and student (and industry) demand.

ACM/IEEE Guidelines. The final main element of the rationale is not to address a particular problem with the existing program, but rather a positive objective: to bring our curriculum in alignment with the guidelines published by our professional societies. Every decade or so, the ACM (Association for Computing Machinery) and IEEE (Institute of Electrical and Electronics Engineers) publish a set of content guidelines for how to construct a well-rounded university-level computer science program. (Longwood's original CS major was designed when the 1978 guidelines were current; they were later updated in 1991 and 2001.) The ACM/IEEE committee on curricula is developing a current version to be published in 2013, and our program revision was designed with these up-to-date guidelines in mind.

Other changes. There are a few changes to the program not explained by the above rationales:

- We remove the explicit requirement that the electives be at the 300-400 level. This had been a way to exclude 121 (our gen ed course) and the 200-level language courses; we decided that if a student takes an additional language course that should count as an elective, and we ruled out 121 explicitly.
- We change the discrete math requirement from CMSC300/MATH300 to the new

course MATH 175, a 2-credit freshman-level course in discrete math. This is in conjunction with a separate proposal to create that course, but in any case we support a move to a lower-level 2-credit course to fill this niche in our major, because the current CMSC 300 is really a mix of several different topics, some of which need to come early and others a bit later in the student's academic development. (Those that should come later have mostly been moved to the new CMSC 208.)

- We combined several 200-level programming language courses into a single course number (280), which will be offered in specific-language variants. This is administratively easier, and it lets us react more quickly to the fast-changing demands of industry.
- We merged the existing CMSC 408 (Software engineering) and CMSC 461 (Seminar in computer science) under the 461 number designated for capstone courses. The former 461 was not as academically rigorous as we desired; and since software engineering requires just about everything in computer science, it would serve well as a rigorous capstone course. The content is largely unchanged from the existing 408, except that assessment (currently an MFT) is added.

APPENDIX A. COURSE SUBSTITUTIONS

After going through the academic plans of all the currently-declared majors, the following set of substitutions appears to cover all our bases.

For students during the transition period who are on an older catalogue:

- Count new-201 for old-301. (same course)
- Count new-208 for old-308. (imperfect substitution, but similar)
- Count new-242 for old-342. (very similar course)
- Count new-262 as 300-level elective. (counts towards 27 hours of electives)
- Count new-280 for old-220. (effectively the same course)

Count new-461 + any-300-400 for old-408 + old-461. (408 no longer offered as such; new 461 basically same as old 408; require elective to keep credit requirement the same)

Require MATH 300 if they have not taken CMSC 300. (old major allows either one)

In general, for students currently taking CMSC 160 or 162, it will be better to elect to the new catalogue next year. However, if they choose not to do so for whatever reason, the above set of substitutions will also work for them.

In some cases, even for students who are currently sophomores (and thus taking CMSC 300, CMSC 301, etc), it may make sense to elect into the new catalogue. There also may be some students who are taking some CMSC courses now but don't declare their major until next year. We won't let them out of taking the substantially new courses, but the following substitutions would be appropriate (courses that are unchanged or whose new versions are simpler/smaller than their old versions).

For students during the transition period who are on the new catalogue but have taken old-catalogue classes:

- Count old-301 for new-201.
- Count old-300 for new-MATH 175.
- Count old-342 for new-242.
- Count old-220 for new-280.