Top Ten List for Making a Great "Major Requirements" Web Page
McKenna Quam, Summer 2023

Introduction

The Center for Inclusive Computing (CIC) at Northeastern recently undertook a research project to study the relationship between the curricular complexity of undergraduate CS degrees (Heileman, 2018) and the representation of women graduating with a degree in CS (paper is under review). To this end, we examined the degree requirements of 75 different U.S. CS departments using the resources publicly available on department and university web pages. In looking at the web pages, our experience mirrors that of the student - whether prospective, current or looking to transfer from another institution. On average, the process of figuring out the degree pathway options took us two hours per school, but varied significantly depending on the accessibility and clarity of the information presented. Based on this experience, we have created this Top Ten List. We hope it is useful to you!.

1.   Course Plan - Have one (or three!)

At a basic level, a course plan shows students how they can progress through the major. On a detailed level, course plans guide students in the creation of *balanced* schedules, helping them avoid overloading any given semester with too many heavy-workload classes. Much of the time, a university's CS department almost always offers *more than* one "archetypal" pathway - but very few present more than one course plan. Indeed, the course plan that is most often absent is the one that accommodates students who are not calc ready and/or are true CS beginners (See example #1). Acknowledging students' different starting points through multiple course plans signals that the department is welcoming to all students, regardless of prior experience. This matters because prior experience is not uniformly distributed with regard to gender,race/ethnicity, and socioeconomic status. (Brodley, 2022)

2.   Hyperlink Everything - Avoid Dead Ends

Best practices here are clear: i) every course in a department's requirements list should be hyperlinked to its description; ii) each description should include all prerequisites; and iii) each prerequisite should be hyperlinked to its description. In other words, avoid dead ends that require students to go searching for the information elsewhere, potentially ending up on old or incorrect pages (See example #2)
.

3.   Words Please - Course Codes Mean Nothing to Me

It takes students a long time to learn the course code lingo. This is particularly true for students who are brand new to university, transferring in from another institution, and/or coming from outside the department. A great practice is to always include both the course code and the name of the course (See example #3).

4.   Quick! What Year is it?

Every page that a student is using to figure out which courses to take should indicate the year of the catalog. If the catalog is archived, *that fact should be noted in* **bold**. This is an easy fix that keeps students from making the mistake of building plans using outdated information (See Example #4).

5.   Link to All Courses - Not Just The CS Ones

CS degrees often require classes in other departments (e.g., math). As such, requirements pages should link easily to the "single source of truth" course catalog (for the appropriate year). Some sites, likely in an effort to be

efficient, offer static pdfs or other summaries of the outside-the-department-required courses. These are dead ends and mean that students have to go looking for that information on their own, running the risk of going to the wrong place and/or building off of outdated information (See example #2).

6.   Gen Eds - Tell Me What I Am Missing (Not What I Have)

By and large, the requirements pages we looked at are pretty good at listing which general education requirements the major satisfies. Less common, but - we would argue - more useful, is a requirements page that tells you what the major does not cover, highlighting the "gaps" the student needs to to satisfy elsewhere. (See example #2).

7.   Organize in Tables

The list of requirements can be organized in many ways. By far the easiest to read is a table format with columns for each of the following: i) hyperlinked course code, ii) the course name, and iii) credit hours (See example #3). This format has many benefits, chief among them that it allows the department to indicate the full set of class options that fulfill a given requirement (e.g., "take one from this list… take two from this list").

8.   Unhide Hidden Requirements

If something is a prerequisite for a required course, then it is a required course and should be listed as such on the requirements page and in course plans (See example #5).

9.   Be Wary of Flow Charts

Because CS is a progressive degree, some schools present flow charts to illustrate the different paths through the degree. However, because there are many fulfillment options for certain courses and because of the heavy dependency on prerequisites, flow charts quickly become unreadable.

10.   Major Requirements - Start at the True Beginning

The requirements list should be organized with the understanding that some students may not be calculus ready and/or may have no CS experience when they enter university. If a school has a CS0 that is required/recommended, then it should be included on the requirements page. Students with CS experience know they have CS experience, but beginner coders might not understand that "Computer Science 1" may not be the first class they should take (See example #5).

Conclusion - Why is a good website so important?

A confusing or incomplete website invariably means that advisors will spend more of their time answering factual questions (what are the prereqs? How many credits do I need to graduate? What is the first course?) and less time on actual advising. A great website means students - current, prospective, transferring - can answer many questions themselves and bring the harder, more interesting questions to their advisor meetings.

Citations

C. E. Brodley, [Expanding the Pipeline: Addressing the distribution of prior experience in CS1 Computing](#) Research News (CRN), June 2022, Vol. 34 No. 6.

Heileman, Gregory L., et al., [Curricular analytics: A framework for quantifying the impact of curricular reforms and pedagogical innovations.](#), 2018, *arXiv preprint arXiv:1811.09676*

Appendix

Example 1

University of Wisconsin Madison provides students with several sample plans catered to students with different levels of Math and Computer Science in addition to the 4-Year plan they have linked in the catalog.
https://www.cs.wisc.edu/undergraduate/sample-four-year-planning-guides/

Example 2

San Jose State University provides the list of non major graduation requirements at the top of the catalog page with each requirement hyperlinked to its respective section on the general graduation requirements page. All of courses on this page are hyperlinked and the prerequisites in their descriptions are too. On the left there is a menu allowing students to access the course descriptions for all other departments.
https://catalog.sjsu.edu/preview_program.php?catoid=14&poid=8384&_gl=1*k7c5d3*_ga*ODQ5MjAwNDM2LjE2ODkyODA1MTg.*_ga_WQPDD4JE9Q*MTY4OTk2NTAzOC4zLjAuMTY4OTk2NTAzOC4wLjAuMA..

Example 3

Northeastern's catalog uses a table format which helps them display their computer science curriculum which has a lot of student choice. The table format allows them to easily display both the course code and course name.
https://catalog.northeastern.edu/undergraduate/computer-information-science/computer-science/bscs/#programrequirementstext

University of California Berkeley also uses a similar table format.
https://guide.berkeley.edu/undergraduate/degree-programs/computer-science/#majorrequirementstext

Example 4

Colorado state lists their catalogue year at the very top of the page in a large font so it is easy to see.
https://catalog.colostate.edu/general-catalog/colleges/natural-sciences/computer-science/computer-science-major/computer-science-concentration/#requirementstext

Example 4

The University of Illinois Urbana-Champaign lists their 1 credit intro course at the top of their requirements while making it clear that it is not a part of the core. Having CS 0 listed on the page also demonstrates that this page is starting from the beginning and that students will not have to worry about missing hidden requirements.
http://catalog.illinois.edu/undergraduate/engineering/computer-science-bs/#degreerequirementstext